L'AMIGA

Immagini,
Suoni e
Animazioni
sul
Commodore
Amiga

Michael Boom

ENDORSED BY AMIGA

IHT GRUPPO EDITORIALE
DIVISIONE LIBRI

COLLANA INFORMATICA

L'AMIGA

L'AMIGA

Immagini,
Suoni e
Animazioni
sul
Commodore
Amiga

Michael Boom

Una pubblicazione
IHT Gruppo Editoriale S.r.l.
Via Monte Napoleone, 9
20121 Milano

Copyright © 1986 by Michael Boom
Copyright © 1987 by IHT Gruppo Editoriale, Milano

Proprietà letteraria riservata. Questo libro non può essere
copiato o fotocopiato, tradotto o ridotto in forma leggibile,
né tutto né in parte, da qualsiasi mezzo elettronico
o meccanico, senza previa autorizzazione scritta dell'Editore

Titolo originale dell'opera:
The Amiga: Images, Sounds, and Animation on the Commodore Amiga
Edizione in lingua inglese:
Microsoft Press, Redmond, WA, USA

Direzione editoriale della collana di Massimiliano Lisa
Traduzione di Corrado Dal Corno
Collaborazione editoriale di Alberto Farina
Impaginazione a cura di Antonio Gaviraghi

Grafica di copertina a cura di Ted Mader and Associates
Grafica dell'interno a cura di Woodfin Design
Illustrazioni di Rick Bourgoin
Per le immagini di Polyscope si ringrazia Dan Silva
Fotografie degli schermi di Nick Gregoric

ISBN 88-7803-000-7

Prima edizione: novembre 1987
Prima ristampa: gennaio 1988

DEDICA

Questo libro è dedicato a tutti coloro che hanno collaborato al progetto Amiga, creando un grande computer e tenendomi molto occupato per un anno e mezzo.

SOMMARIO

Prefazione		*XI*
Ringraziamenti		*XIII*
Introduzione		*XV*
SEZIONE 1	**LA MACCHINA**	
Primo capitolo	L'Amiga visto da vicino	*3*
SEZIONE 2	**IMMAGINI**	
Secondo capitolo	Nozioni fondamentali di videografica	*33*
Terzo capitolo	Software e hardware grafico per Amiga	*61*
Quarto capitolo	La grafica e l'Amiga BASIC: schermi, finestre e palette	*95*
Quinto capitolo	La grafica e l'Amiga BASIC: la creazione di immagini	*125*
Sesto capitolo	La grafica e l'Amiga BASIC: miscellanea	*159*
SEZIONE 3	**SUONI**	
Settimo capitolo	Nozioni fondamentali di musica elettronica	*189*
Ottavo capitolo	Software e hardware musicale per Amiga	*213*
Nono capitolo	Il suono e l'Amiga BASIC: musica e sintesi vocale	*239*
SEZIONE 4	**ANIMAZIONI**	
Decimo capitolo	Nozioni fondamentali di animazione	*271*
Undicesimo capitolo	Software e hardware dell'Amiga per l'animazione	*289*
Dodicesimo capitolo	L'animazione e l'Amiga BASIC: creazione di oggetti in movimento	*317*
Tredicesimo capitolo	L'animazione e l'Amiga BASIC: come controllare il movimento	*351*
Conclusione	Il futuro: le possibiltà creative dell'Amiga	*375*
Appendice A	Sintassi dei comandi in Amiga BASIC	*378*
Appendice B	Case produttrici citate nel testo	*385*
Indice analitico		*387*

PREFAZIONE ALL'*AMIGA*

I media elettronici diversificano il nostro mondo consentendoci di accedere quasi simultaneamente a una molteplicità di esperienze. Nel giro di poche ore possiamo ascoltare un disco dei Beatles, parlare con un amico che vive in un altro continente, guardare il Super Bowl, persino trovarci a Vienna con Mozart e Salieri.

In quasi tutti i casi, la popolarità e l'impatto culturale di uno strumento elettronico sono andati di pari passo con il suo livello di fedeltà nella riproduzione di immagini e suoni: i migliori prodotti dell'elettronica sono quelli che mettono in una scatola la vita stessa. Raramente i personal computer hanno potuto competere con altri apparecchi elettronici nel campo della riproduzione realistica di suoni e immagini, ma la dimensione audiovisiva dell'Amiga è del tutto nuova per il mondo dei computer, tradizionalmente considerati puri divoratori di numeri. Marshall McLuhan ha detto: "Il mezzo di comunicazione è il messaggio". E il messaggio dell'Amiga è audiovisivo.

L'Amiga è il primo personal computer che si avvicina alla qualità delle immagini di un televisore e alla fedeltà del suono di un apparecchio stereo. Ma, a differenza della televisione e dello stereo, l'Amiga è un mezzo interattivo. Con la televisione lo spettatore vede un altro mondo riprodotto sullo schermo; con l'Amiga si può entrare nello schermo ed essere di persona nel mondo riprodotto. Con lo stereo si ascolta la musica scritta e suonata da altri; con l'Amiga è possibile modificare quella stessa musica o crearne una propria. Si tratta di qualcosa che va al di là del vedere o dell'ascoltare: l'Amiga è un mezzo creativo.

Questo strumento creativo è arrivato proprio al momento giusto. Negli ultimi venticinque anni, mentre intere generazioni crescevano davanti allo "stupido tubo", il dottor Marian Diamond dell'Università della California ha effettuato accurate ricerche sul cervello umano. I risultati sono stati catalogati e il dottor Diamond ha dimostrato che il miglior modo per mantenere viva l'intelligenza è la continua interazione con l'ambiente che ci circonda. In poche parole, "usa l'intelligenza o la perderai". Oppure, come dice un proverbio cinese: "Sento e dimentico, vedo e ricordo, faccio e capisco".

L'apprendimento necessita naturalmente di motivazioni. McLuhan deve aver avuto una visione dell'Amiga quando ha detto: "Chi separa l'educazione dal divertimento non ha capito niente né di una, né dell'altro". Il bello dell'Amiga è che si può apprendere in modo attivo, piacevolmente intrattenuti da effetti audio e video incredibilmente realistici. Con l'Amiga, il nostro istinto naturale di creare, sperimentare ed esplorare viene arricchito di un nuovo vigore.

I videogame di alcuni anni fa riproducevano immagini e suoni primitivi, erano anche interattivi, ma si rivelarono poi una moda passeggera, perché mancavano di realismo audiovisivo e perché i

contenuti non potevano interessare a lungo. Ora sappiamo che il software di alta qualità deve essere semplice da capire, "vivo" sotto il profilo audiovisivo per attirare e catalizzare la nostra attenzione, nonché serio e pertinente quel tanto che basta a tenere occupato il cervello. Grazie alla semplicità delle operazioni e alle sue enormi potenzialità, l'Amiga è il mezzo ideale per gli artisti del software che vogliono ampliare i confini del settore. Di fronte a un software creato utilizzando tutto il potenziale dell'Amiga, molte menti trasaliranno e molte persone resteranno estasiate.

La Western Union ebbe la possibilità di ottenere l'esclusiva del telefono, ma se la lasciò sfuggire, non avendo intuito il grande avvenire di quell'apparecchio. Sulle prime si pensava anche che la televisione fosse adatta solo per comunicazioni dedicate alla protezione civile. Solo il tempo e le menti creative sapranno dire qual è il vero potenziale dell'Amiga.

Recentemente è stato fatto molto nel settore del "desktop publishing", che, migliorando la qualità e le opportunità di controllo da parte dell'utente, potrebbe rivoluzionare l'intero settore della comunicazione su carta stampata. Senza sottovalutare l'importanza della cellulosa, l'audiovisivo incombe come mezzo alternativo per la trasmissione di messaggi. È quindi ragionevole supporre che, attorno all'Amiga, si venga a creare un mercato di "video da tavolo" che, alla fine, abbia il sopravvento sul "desktop publishing".

Quale primo computer audiovisivo, l'Amiga potrebbe essere l'innovazione in grado di innescare una rivoluzione intellettuale, come è successo con l'invenzione della stampa di Gutenberg. "*L'Amiga: Immagini, Suoni e Animazioni*", scritto con molta cura da Michael Boom, vi offre l'opportunità di partecipare a questa rivoluzione. Viva l'Amiga!

Trip Hawkins
Presidente della Electronic Arts

RINGRAZIAMENTI

Non avrei mai potuto scrivere un libro come questo, che tratta di argomenti così diversi, senza la collaborazione di altre persone. Vorrei ringraziare (senza alcun ordine di merito) tutti i progettisti della Commodore-Amiga, che mi hanno insegnato una serie di nozioni utilissime riguardanti la macchina da loro sviluppata: Sam Dicker, Carl Sassenrath, Bill Kolb, David Needle, Howard Stolz, Rick Geiger, Mitchell Gass, Stan Shepard, Rick Rice, Caryn Havis, Neil Katin, Jack Haeger, R. J. Mical, Barry Whitebook, Terry Ishida, Bob Pariseau, e tutti coloro che hanno risposto alle mie domande nonostante fossero sommersi dagli impegni della produzione. Vorrei ringraziare David Grady della Electronic Arts per tutte le informazioni su Deluxe Paint, e Stewart Bonn per avermi fornito le copie dei primi esemplari di Deluxe Video, per avermi mostrato Instant Music e per avermi permesso di infastidirlo con le mie richieste di alcuni fantastici videogiochi della Electronic Arts.

Molte società mi hanno prestato le apparecchiature da utilizzare durante la stesura di questo libro, un grande aiuto per uno scrittore con un budget limitato. Vorrei ringraziare Jerry Kovarsky della Casio, Bill Mohrhoff della Teac, e in particolare Jeff Stenehjem della Xerox per il tempo e le informazioni che mi hanno concesso, al di là del comune senso del dovere.

Naturalmente la ricerca è la parte più divertente del lavoro. Il compito più arduo è invece ordinare in forma scritta ciò che si è appreso e ritoccarlo con cura per la pubblicazione. Vorrei ringraziare il personale, molto preparato, della Microsoft Press, per avermi assistito in tutte le fasi della preparazione del libro, in modo particolare Dave Rygmyr per aver sostenuto spiritualmente il manoscritto dalla prima all'ultima pagina, con incrollabile buonumore, abilità e interesse nel contenuto; ringrazio anche Allan McDaniel e Tom Corbett per le risposte ai miei problemi dell'ultimo minuto sull'Amiga BASIC.

Infine vorrei ringraziare Lynn Morton per il sostegno e la comprensione che mi ha dimostrato anche nei momenti di ossessione, esasperazione e stanchezza. Grazie, tesoro, ormai è fatta. Possiamo rilassarci.

INTRODUZIONE

Se chiedete a qualcuno quale sia il suo mezzo di espressione artistica preferito, ci sono poche possibilità che la risposta sia "il personal computer". I più ritengono che i personal computer siano adatti a gestire un database o un foglio elettronico, o per elaborare testi molto brevi; ma non pensano al computer come a un mezzo espressivo. E quando lo fanno, lo mettono su un piano addirittura inferiore ai gessetti e ai pifferi in quanto a qualità di immagini e suoni.

Chi usa l'Amiga sa che questi preconcetti sono infondati. Grazie all'hardware e al software creati appositamente per la grafica e per la musica, l'Amiga utilizza colori e suoni con la stessa facilità con cui gestisce un foglio elettronico ed elabora delle parole. Adoperando il software altamente creativo disponibile per l'Amiga, l'utente può creare immagini dai colori squisiti e dai dettagli raffinati, nonché produrre musica ed effetti sonori che potrebbero competere con un sintetizzatore professionale. Con l'Amiga è possibile combinare tra loro capolavori figurativi e musicali per ottenere animazioni di qualità paragonabile ai cartoni animati, un'attività che finora è stata accessibile solo agli animatori professionisti, gli unici ad avere accesso a costosissime apparecchiature.

Come per qualsiasi strumento, l'uso creativo dell'Amiga richiede studio e pratica. Questo libro è stato scritto per fornirvi entrambe le cose, attraverso l'insegnamento dei concetti necessari a capire ciò che bisogna fare e attraverso una serie di esempi che vi permettono di sperimentare quello che state imparando. I principianti troveranno i concetti fondamentali della riproduzione grafica e sonora su computer in appositi capitoli all'inizio delle rispettive sezioni. Una volta acquisite queste nozioni si può tranquillamente passare ai capitoli successivi, che illustrano l'uso di programmi quali Deluxe Paint, Deluxe Music e Deluxe Video. Esempi accuratamente dettagliati forniscono la necessaria esperienza pratica nell'uso dei programmi.

Anche gli utenti più esperti possono trovare materiale di loro interesse. I capitoli dedicati ai programmi sopracitati spiegano concetti più avanzati e rivelano alcuni trucchi che consentono di ottenere risultati unici. Questi capitoli si occupano anche di altro software disponibile sul mercato e spiegano come trasformare l'Amiga in uno strumento ancora più completo, grazie all'aggiunta di diversi elementi hardware. Anche i programmatori in linguaggio BASIC hanno di che sbizzarrirsi nei capitoli riservati agli intricati dettagli sui comandi grafici e sonori dell'Amiga BASIC; inoltre sono stati inseriti alcuni programmi dimostrativi che insegnano a fondere immagini e suoni in modo da ottenere risultati di eccellente qualità.

LA STRUTTURA DEL LIBRO

Il testo è suddiviso in quattro sezioni per facilitare il lettore che deve riferirsi a una specifica area di interesse. La sezione 1 è dedicata alla presentazione dell'Amiga, un'analisi dettagliata dell'hardware e del software sistema che rivela ciò che fa grande questo computer. La sezione 2 è riservata alla grafica, alle nozioni necessarie a creare immagini ferme sullo schermo. Il suono viene ampiamente trattato nella sezione 3, come fare della musica, simulare una voce o produrre effetti sonori utilizzando la sintesi interna dell'Amiga. La sezione 4 rivela i segreti dell'animazione, spiegando le tecniche che consentono all'utente di "dare vita" alle figure disegnate sullo schermo.

Le ultime tre sezioni sono strutturate allo stesso modo. Il primo capitolo di ogni sezione è una sorta di manuale che introduce i concetti fondamentali trattati nella sezione stessa. Ad esempio, nella sezione 2 viene descritto il modo in cui i monitor riproducono le immagini, nella sezione 3 il modo in cui l'orecchio umano percepisce i suoni, nella sezione 4 le modifiche da apportare alle immagini ferme per creare l'illusione del movimento. Il secondo capitolo di ogni sezione illustra il funzionamento di uno specifico programma applicativo – Deluxe Paint nella sezione 2, Deluxe Music nella sezione 3, Deluxe Video nella sezione 4 – e si occupa anche di hardware e software addizionali disponibili per l'Amiga. I restanti capitoli di ogni sezione sono dedicati ai comandi e alle funzioni dell'Amiga BASIC.

COME USARE IL TESTO

La prima sezione di questo volume è stata scritta per consentire la piena comprensione delle componenti dell'Amiga, in modo da evitare confusioni nelle tre sezioni seguenti. Se conoscete già bene l'Amiga è possibile passare direttamente alla sezione seguente che più vi interessa.

All'interno di ogni sezione è consigliabile leggere il primo capitolo, che introduce i concetti fondamentali trattati negli altri capitoli della stessa sezione. Una volta letto il primo capitolo, si può leggere il secondo e saltare i capitoli dedicati al BASIC (se siete più interessati ai programmi che alla programmazione) oppure potete saltare il secondo capitolo se ciò che vi interessa maggiormente è il BASIC. Se non avete mai provato ad usare l'Amiga BASIC è consigliabile leggere il quarto capitolo (nella sezione 2), che vi dice come inserire un programma e farlo girare.

Può capitare che qualcuno voglia provare a leggere i capitoli sul BASIC pur non essendo un programmatore. In effetti, molte delle informazioni fornite da quei capitoli sono utili ai non programmatori, e, leggendo prima il quarto capitolo, ne saprete abbastanza per caricare e far girare i programmi dimostrativi, pur senza capire come funzionano. Per favorire la digitazione dei comandi e diminuire la

possibilità di errori, i programmi dimostrativi (tranne alcune eccezioni) sono semplici e brevi. Ricordatevi solo di premere il tasto RETURN alla fine di ogni riga per far sì che l'Amiga la memorizzi.

Il libro è provvisto di due appendici molto utili. L'appendice A contiene una lista di tutti i comandi e le funzioni BASIC citati nel testo, seguiti dalla sintassi dell'istruzione e da una breve spiegazione. Quando scriverete programmi vostri, questa appendice costituirà un ottimo punto di riferimento nel caso vi sfugga la giusta sintassi di un comando, o nel caso non vi ricordiate quale comando adempia a una certa funzione. L'appendice B contiene un elenco di tutte le società produttrici di hardware e software menzionate nel libro, con gli indirizzi a cui potete scrivere per avere informazioni sui prodotti. Le appendici sono seguite da un indice molto completo, che facilita la ricerca di qualsiasi argomento trattato nel corso del libro.

COSA SERVE PER METTERE IN PRATICA GLI ESEMPI DEL LIBRO

Per mettere in pratica gli esempi esposti, l'utente deve avere a disposizione quattro diversi package di software: Deluxe Paint per il terzo capitolo, Deluxe Music per il quarto capitolo, Deluxe Video per l'undicesimo capitolo e naturalmente l'Amiga BASIC per tutti i capitoli dedicati al BASIC. Quest'ultimo viene venduto unitamente alla macchina, gli altri sono reperibili presso tutti i rivenditori di prodotti Commodore.

Per far girare gli esempi è sufficiente un'attrezzatura minima. Nonostante in alcuni casi (quando viene spiegato come registrare le vostre creazioni) servano una macchina fotografica, una stampante, un videoregistratore o un registratore a cassette, di solito non serve altro che l'Amiga, il monitor e il mouse.

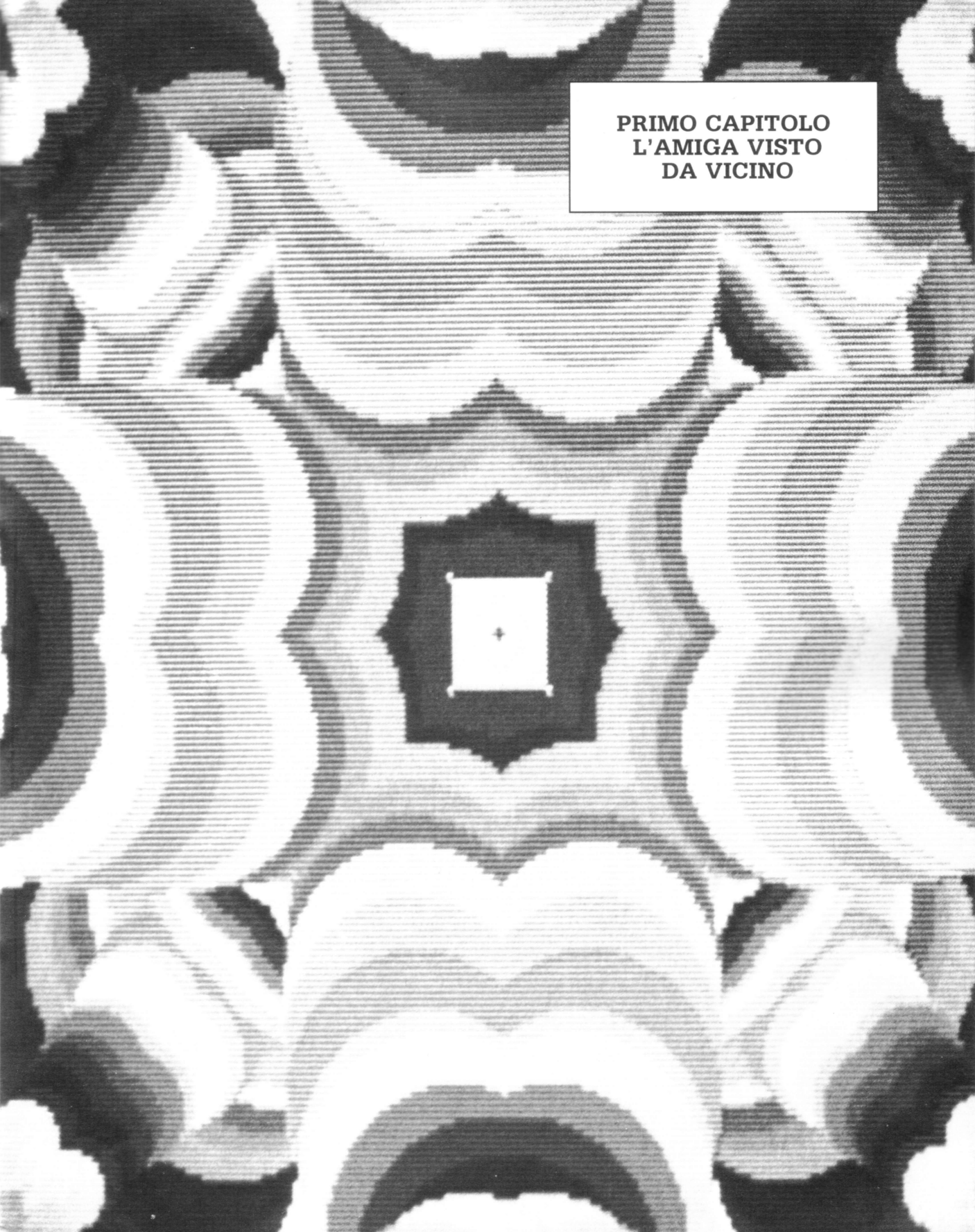
È a questo punto importante sottolineare che se siete un principiante e non sapete come usare il mouse, i menu, la tastiera o altro, è consigliabile leggere *Introduzione all'Amiga*, un manuale compreso nella confezione del computer, che descrive le nozioni fondamentali per poter usare la macchina. Una volta appresi questi concetti non dovrebbero insorgere problemi di alcun genere.

Le ore che trascorrerete utilizzando in modo creativo la grafica e il suono dell'Amiga potranno essere i momenti più divertenti che abbiate mai passato davanti ad un computer. Quello che imparerete grazie a questo libro vi aiuterà a trasformare le vostre idee in risultati audiovisivi concreti con una fatica trascurabile. Gli amici rideranno forse di voi quando vi siederete alla tastiera, ma quando inizierete a far correre il mouse...

SEZIONE 1

La macchina

Questa sezione è dedicata a un'analisi accurata dell'Amiga e delle sue capacità. La presentazione della strumentazione Amiga necessaria al funzionamento e alla riproduzione grafica e sonora è seguita dalla trattazione dei due diversi livelli di software di cui dispongono tutti gli Amiga. Le informazioni e i principi esposti in questa sezione svelano alcuni segreti delle prestazioni dell'Amiga e forniscono all'utente una solida conoscenza di base per affrontare in tutta tranquillità le sezioni successive.



PRIMO CAPITOLO
L'AMIGA VISTO DA VICINO

Trovandosi per la prima volta di fronte a un Amiga, si vede solo metà di ciò che lo fa funzionare. La metà che potete vedere e toccare con mano – la tastiera, l'unità centrale, il monitor e il disk drive – costituisce l'hardware. Invece, la metà invisibile dell'Amiga costituisce il software.

Il software è un insieme di istruzioni, immagazzinate nel computer o su floppy disk, che determina l'interazione delle varie componenti e funzioni, ad esempio l'invio di messaggi dalla tastiera al computer e la visualizzazione di immagini sullo schermo del monitor. Il software consente al computer di svolgere compiti utili all'operatore. Senza software, il computer sarebbe un soprammobile inutile e costoso.

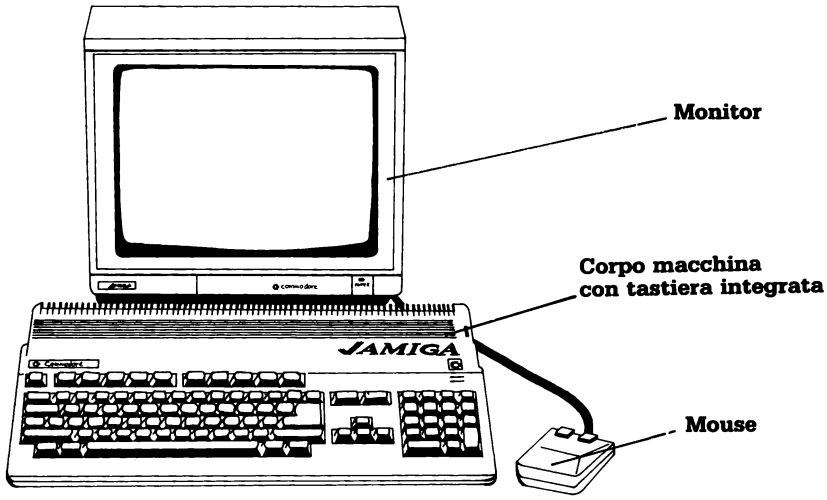
Un sistema Amiga completo è l'unione di due unità cooperanti: l'hardware e il software. Per comprendere l'intero sistema, è necessario che conosciate entrambe le metà del computer.

L'HARDWARE

Le Figure 1-1/b-c mostrano le quattro componenti principali di un sistema Amiga di base: l'unità centrale, il monitor, la tastiera e il mouse. All'Amiga si possono collegare altre componenti utili, dette periferiche, quali ad esempio le stampanti, i modem o disk drive addizionali, finché il sistema non si espande oltre la scrivania e invade l'intera stanza. Tuttavia, per i meno esigenti, queste quattro componenti essenziali sono sufficienti per rendere operante il sistema.

Figura 1-1.

Le tre componenti essenziali di un sistema Amiga 500.



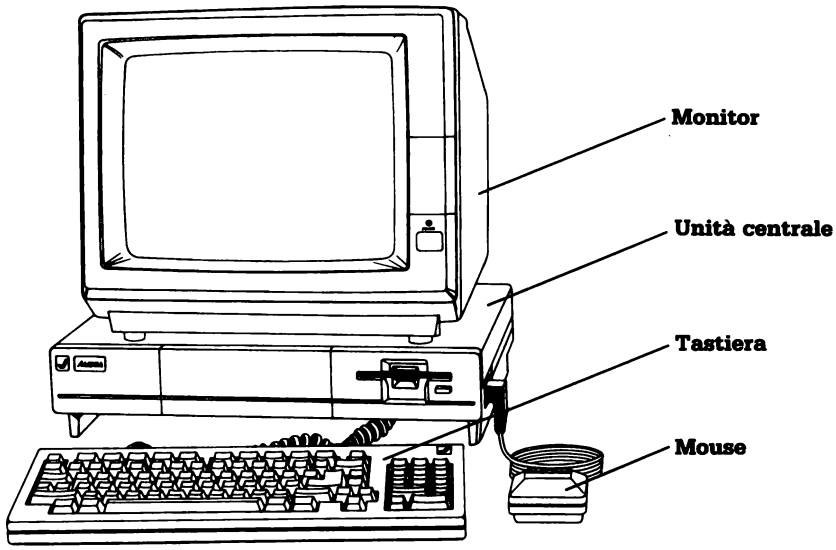


Figura 1-1/b.

Le quattro componenti essenziali di un sistema Amiga 1000.

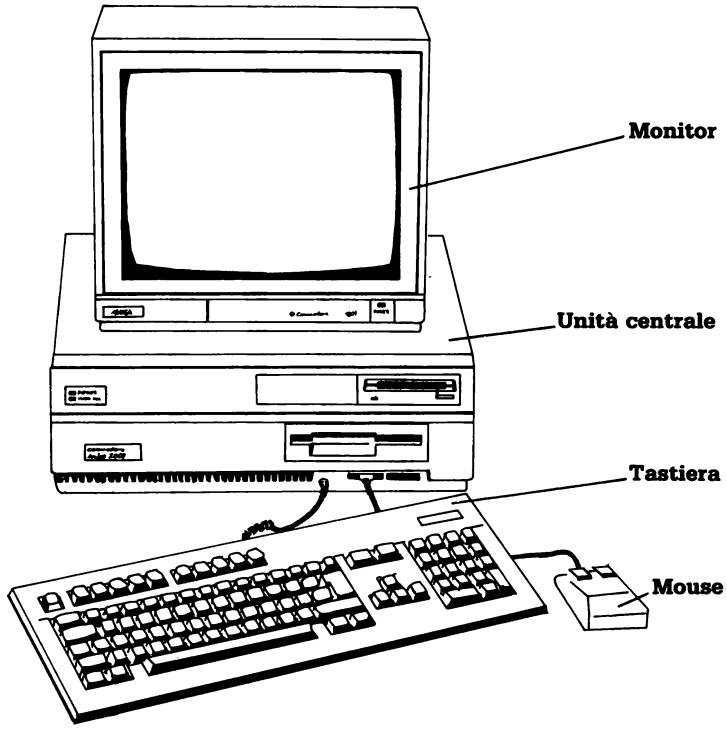


Figura 1-1/c.

Le quattro componenti essenziali di un sistema Amiga 2000.

IL MONITOR

Probabilmente il monitor è il primo componente del sistema Amiga a catturare l'attenzione del suo utente. Il monitor rappresenta il canale di comunicazione tra l'operatore e il sistema Amiga; fornisce informazione visiva attraverso lo schermo e informazione audio attraverso l'altoparlante.

Al sistema Amiga possono essere collegati tre tipi di monitor differenti: i monitor RGB, i monitor videocompositi e i normali apparecchi televisivi. I monitor RGB (abbreviazione di Red, Green, Blue, vale a dire Rosso, Verde, Blu) presentano un'immagine più nitida e con colori più vivi e visualizzano in modo chiaro anche testi a 80 colonne. Questo tipo di monitor è vivamente raccomandabile a chi possiede un Amiga; infatti, pur comportando una certa spesa, è l'unico monitor che renda pienamente giustizia alle capacità grafiche dell'Amiga.

Il monitor videocomposito è di qualità intermedia fra un apparecchio televisivo e uno RGB. Benché non nitida come quella di un monitor RGB, l'immagine di un monitor videocomposito collegato all'Amiga risulta essere, grazie all'ottimo segnale videocomposito della macchina, di discreta qualità.

La qualità di un comune apparecchio televisivo può variare dall'accettabile al pessimo. Il vantaggio è naturalmente quello di poter utilizzare un normale televisore senza dover acquistare un apposito monitor per computer. Se come monitor usate un televisore a colori, l'Amiga, che deve essere dotato di un apposito modulatore, emette un eccellente segnale TV per la visualizzazione dei colori. Sullo schermo di un apparecchio televisivo di buona qualità l'Amiga può visualizzare fino a 60 colonne di testo leggibile.

Per collegare i monitor, l'Amiga 1000 dispone di tre diversi connettori (detti porte), uno per ogni tipo di monitor. Dal momento che tutte e tre le porte inviano gli stessi segnali video (benché sotto forma diversa), è possibile collegare tre monitor differenti contemporaneamente e visualizzare le immagini video dell'Amiga su tre schermi simultaneamente.

Sia l'Amiga 500 (che possiede solo l'uscita RGB e quella monocromatica) sia il 2000 (che dispone solo dell'uscita RGB) possono essere collegati ad un apparecchio televisivo tramite l'impiego di un apposito modulatore video opzionale.

Oltre ai tre monitor esaminati fino a questo punto, ne è stato realizzato anche un quarto, espressamente dedicato a risolvere il problema della vibrazione dello schermo nel modo "interlace". Questo modello, chiamato "ad alta persistenza", consente di utilizzare i 640x400 (640x512 nel sistema televisivo europeo) pixel anche per la visualizzazione di testi ed è l'unico che permetta di sfruttare completamente le capacità grafiche della macchina. L'unico problema può essere quello del prezzo non troppo accessibile.

LA TASTIERA

La tastiera viene utilizzata per inserire informazioni, parole o numeri nell'Amiga, o per richiedere l'esecuzione di certe funzioni. I tasti sono disposti come su una normale macchina da scrivere.

Troverete anche dei tasti diversi da quelli della macchina da scrivere. A sinistra e a destra della barra spaziatrice sono collocati i due tasti Amiga (uno solo nell'A500), contrassegnati da una "A" rossa. Premendo contemporaneamente un tasto Amiga e un altro tasto, è possibile eseguire operazioni particolari, ad esempio selezionare un'opzione di un menu.

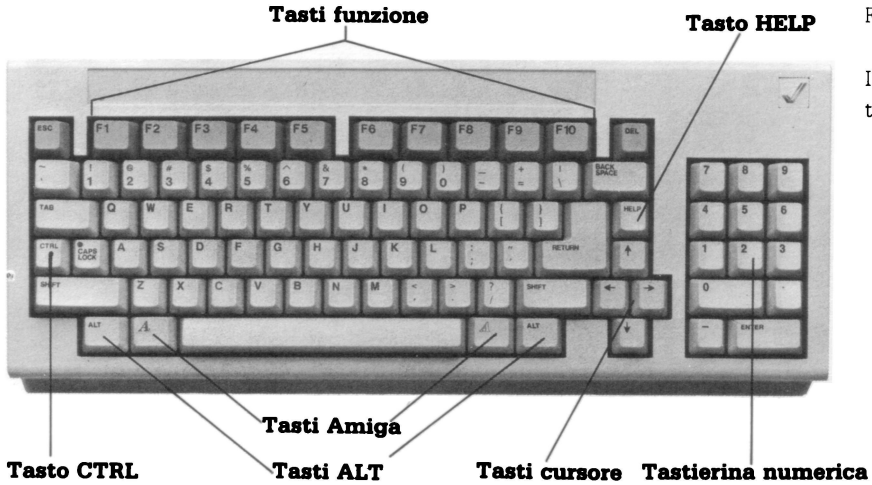


Figura 1-2.

I tasti speciali della tastiera dell'A1000.

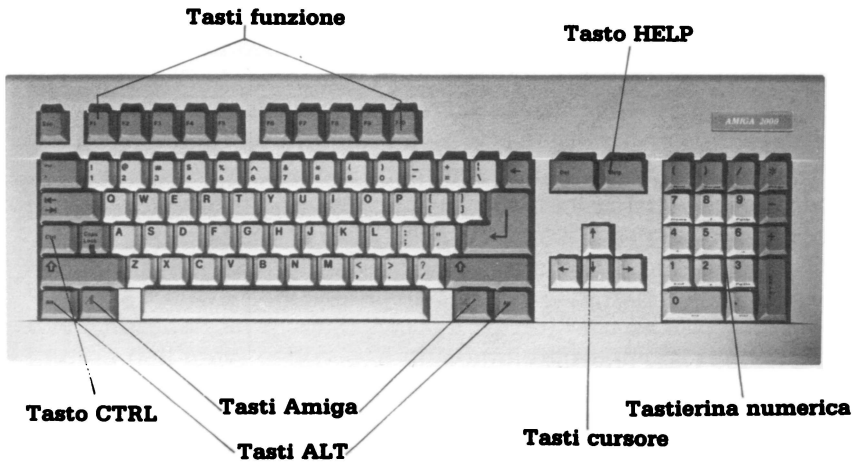


Figura 1-2/b.

I tasti speciali della tastiera dell'A2000.

Accanto ai tasti Amiga sono collocati i tasti ALT, e sopra al tasto SHIFT sinistro troverete il tasto CTRL. I tasti ALT e CTRL sono tasti speciali che, come i tasti Amiga, vengono usati insieme ad altri per eseguire funzioni di vario genere.

Sulla destra della tastiera sono collocati quattro tasti cursore, contrassegnati da frecce, grazie ai quali è possibile spostare il cursore sullo schermo in programmi come quelli di elaborazione testi. Sopra i tasti cursore è collocato il tasto HELP, che potete usare per ottenere informazioni utili sul modo in cui usare un programma. Nella fila superiore della tastiera si trovano allineati dieci tasti funzione (F1-F10), che possono svolgere da soli determinati compiti. La tastierina numerica è situata sulla destra; in tutto simile a una calcolatrice, facilita l'immissione di numeri. Tutti questi tasti speciali sono illustrati nella Figura 1-2.

La tastiera dell'Amiga 500 e del 2000 contiene alcuni tasti in più rispetto a quella dell'A1000; la loro funzione è di gestire la compatibilità IBM (fornita nel 2000 attraverso apposite schede di espansione).

La tastiera (nei modelli A1000 e 2000) è collegata all'Amiga per mezzo di un cavo estendibile a spirale, che permette di spostarla nella posizione preferita, un vantaggio per chi ama scrivere sdraiato o in posizioni particolari. Sul retro la tastiera è dotata di piedini reclinabili che consentono, nel modello A1000, di riporla sotto l'unità centrale.

IL MOUSE

Il mouse consente di limitare l'uso della tastiera ed è l'elemento che più concorre a rendere semplice l'uso dell'Amiga. Muovendo il mouse su una superficie piana si aziona un indicatore (puntatore) sullo schermo. Il mouse può essere utilizzato, ad esempio, per muovere velocemente il cursore nei programmi di elaborazione testi (word processor), per selezionare le opzioni fornite da un menu, oppure per indicare gli oggetti raffigurati sullo schermo che avviano procedimenti particolari come la stampa o il caricamento da disco di sezioni di testo. Il mouse è anche un eccellente strumento grafico: l'operatore può utilizzarlo nei programmi grafici per disegnare figure sullo schermo.

L'UNITÀ CENTRALE

L'unità centrale è il vero e proprio cervello del sistema. In essa sono contenuti tutti i microchip e i componenti elettronici che svolgono il compito di elaborare e immagazzinare i dati. L'unità centrale contiene anche il disk drive, che permette all'utente di immagazzinare dati e programmi memorizzandoli su floppy disk. Esternamente, l'unità centrale ha un aspetto semplice e innocente. Ma guardando al suo interno si capisce

come, in realtà, essa sia un luogo di attività frenetica.

L'interno dell'unità centrale non è un groviglio di fili intricati e lucine colorate (come si potrebbe aspettare chi ha visto troppi film di fantascienza). Al contrario, è una serie ordinata di microchip e di altri piccoli componenti elettronici, allineati su una scheda verde, chiamata "scheda madre". La scheda madre e gli altri componenti installati nell'unità centrale sono indicati nella Figura 1-3.

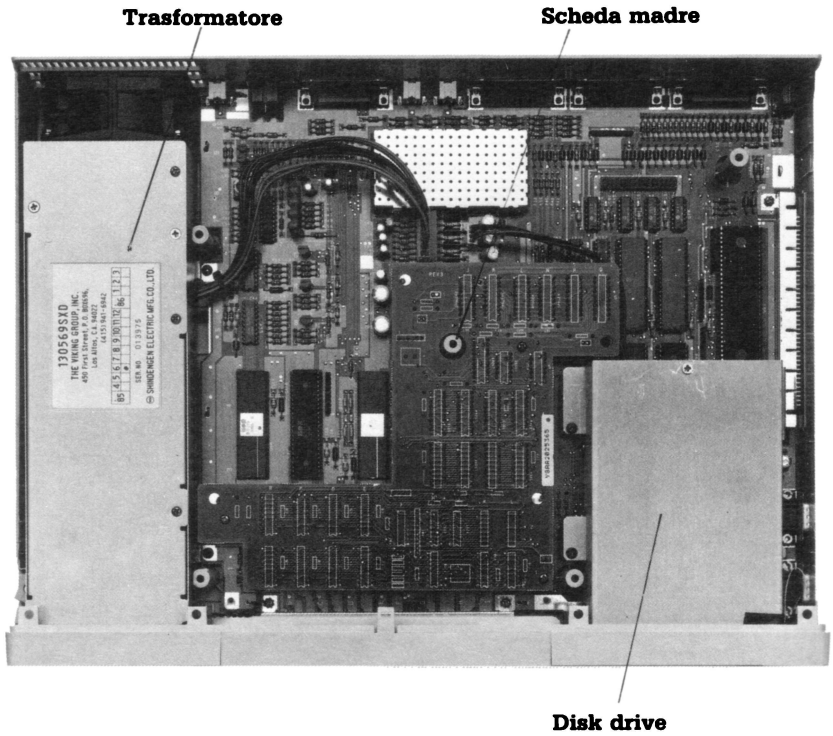


Figura 1-3.

L'interno dell'unità centrale dell'Amiga: la scheda madre, il disk drive interno e il trasformatore.

Il trasformatore

Il trasformatore (che nell'A500 è esterno) svolge la semplice funzione di convertire l'elettricità erogata dalla presa nei bassi voltaggi adatti al funzionamento dell'Amiga. Un'apposita ventola (presente nei modelli A1000 e A2000) raffredda la parte interna dell'unità centrale; è un compito fondamentale, perché quando il computer è acceso e i suoi circuiti sono percorsi dagli elettroni, si genera un calore molto intenso. Un eccesso di calore può falsare il funzionamento dei chip e magari anche danneggiarli. A evitare pericolosi surriscaldamenti contribuiscono le prese d'aria situate sul fondo e sul retro dell'unità centrale. Fate attenzione a non ostruirle mai.

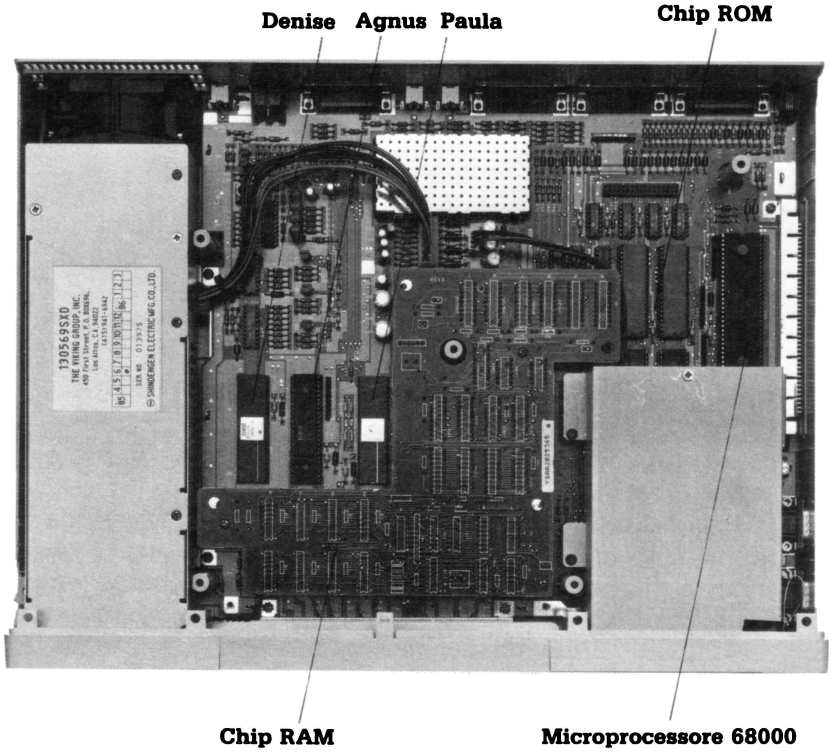
La scheda madre

Gran parte del lavoro svolto dall'Amiga è principalmente opera di quattro chip presenti sulla scheda madre: il microprocessore e i tre coprocessori (vedi Figura 1-4). Il microprocessore è il chip centrale che svolge la maggior parte delle funzioni di "cervello" dell'Amiga, come aggiungere, dividere e inviare comandi e informazioni agli altri chip. I coprocessori, caratteristica esclusiva dell'Amiga e fonte principale delle sue potenti capacità operative, servono da sostegno al microprocessore, occupandosi di funzioni secondarie quali la visualizzazione di immagini e la generazione di suoni. Il microprocessore, alleggerito di questi compiti, è libero di lavorare in piena efficienza.

I tre coprocessori sono chiamati Agnus, Denise e Paula. Il microprocessore ha il nome più altisonante di Motorola 68000.

Figura 1-4.

I chip della scheda madre dell'A1000: il microprocessore, i coprocessori e i chip della RAM e della ROM.



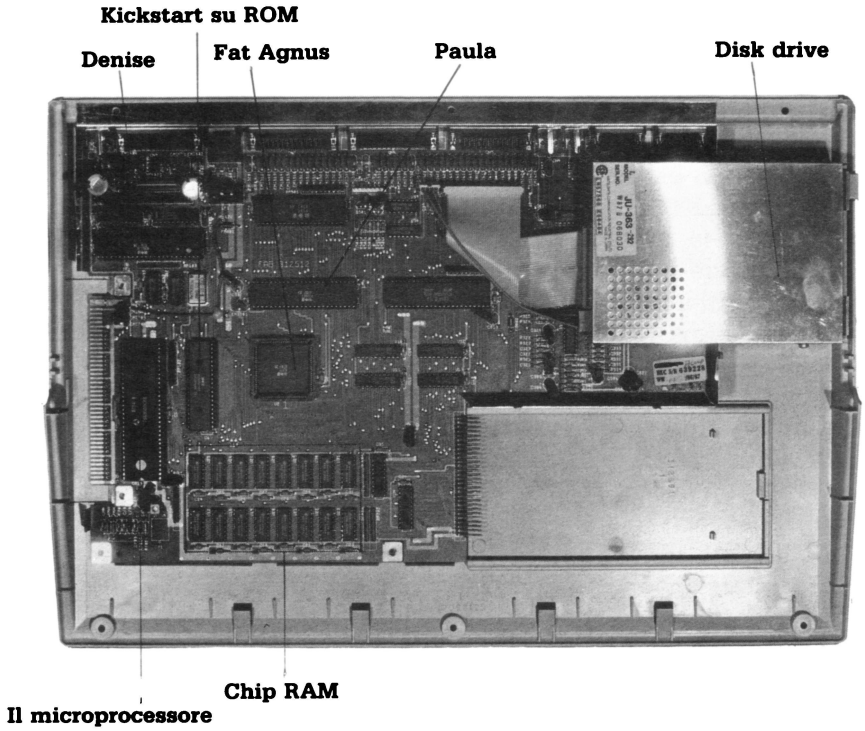


Figura 1-4/b.

I chip della scheda madre dell'A500: il microprocessore, i coprocessori e i chip della RAM e della ROM. Agnus, nell'Amiga 500, è chiamato "Fat" Agnus, perché integra al suo interno più funzioni che sull'A1000. Nella figura è indicato anche il disk drive.

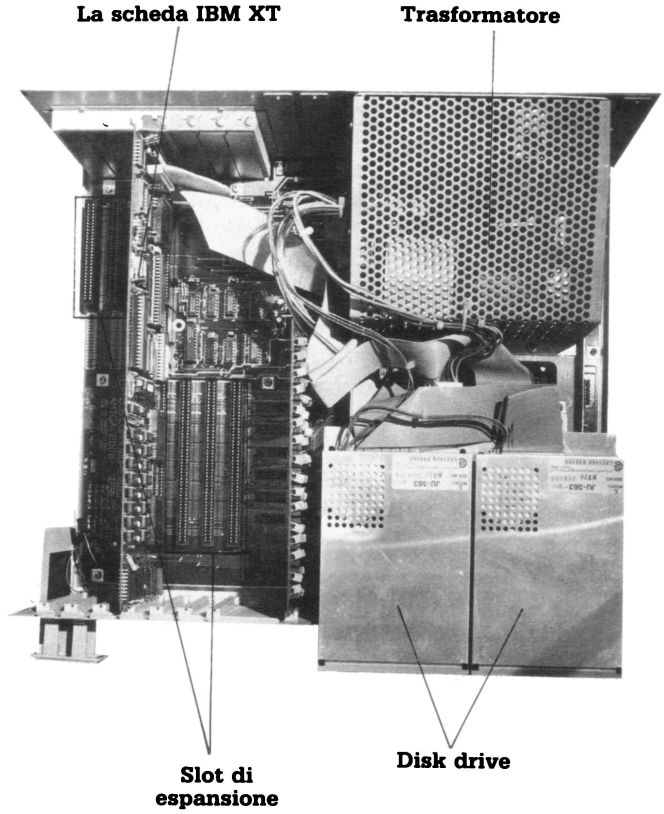


Figura 1-4/c.
L'interno dell'A2000: il trasformatore, i disk drive, gli slot di espansione e la scheda di emulazione IBM XT

Il microprocessore Motorola 68000

La potenza di un microprocessore viene misurata in base alla quantità di dati che può gestire e alla velocità a cui opera. Rispetto ai microprocessori presenti in altri personal computer, il Motorola 68000 16/32 bit è decisamente veloce. 16/32 indica la capacità di elaborazione dei dati: il 68000 elabora dati in gruppi di 16 e 32 bit. Internamente il 68000 lavora su 32 bit di dati alla volta, mentre poi invia al resto del sistema dati a 16 bit. Ciò rende il 68000 più veloce di qualsiasi altro microprocessore a 16 bit, ad esempio quello usato dal PC IBM, e fa dell'Amiga una macchina davvero veloce.

Anche altri computer, come il Macintosh della Apple o l'Atari ST, sono dotati del microprocessore 68000. Ma quello che distingue l'Amiga è il sistema di coprocessori, grazie ai quali la macchina risulta più veloce e sofisticata.

I coprocessori

Ogni coprocessore ha una funzione precisa. Denise è il chip di visualizzazione; controlla il modo in cui le immagini vengono riprodotte sullo schermo. Agnus è il chip di animazione; disegna le figure e le fa muovere sul monitor. Paula è il chip del suono e delle periferiche; genera i suoni dell'Amiga e scambia i dati tra la scheda madre e le periferiche, come il disk drive, il mouse e la tastiera.

I coprocessori sono subordinati al microprocessore centrale e consentono una perfetta sinergia operativa. Infatti, svolgono le loro funzioni senza disturbare il 68000, che è così libero di concentrarsi sulle questioni fondamentali, quali possono essere i calcoli matematici. Questa è una delle caratteristiche più rivoluzionarie dell'Amiga. Molti computer non sono infatti dotati di coprocessori, e il microprocessore è costretto a occuparsi del video, del suono e delle periferiche, con un conseguente rallentamento generale delle operazioni.

Un'altra peculiarità dell'Amiga è costituita dal tipo di relazione che intercorre fra il microprocessore e i suoi coprocessori. Quasi tutti i computer dotati di coprocessori prevedono in ogni caso che questi siano tenuti costantemente sotto controllo dal microprocessore, causando ancora una volta operazioni più lente, anche se non ai livelli dei computer senza coprocessori. Nell'Amiga, invece, il microprocessore funziona diversamente; il sistema è stato progettato in modo che i coprocessori inviino un segnale al 68000 ogniqualvolta hanno bisogno di istruzioni; il microprocessore si dice così: "interrupt driven". In questo modo il microprocessore non è costretto a controllare continuamente ogni chip (procedimento di polling).

È questa la particolarità del microprocessore che più contribuisce alla velocità del sistema. Grazie a questo procedimento operativo e alla collaborazione dei coprocessori, l'Amiga può

lavorare con la velocità necessaria a far muovere le immagini sullo schermo, trasferire rapidamente dati da un disco alla memoria centrale o eseguire calcoli complicati senza far aspettare troppo il suo utente. La presenza dei coprocessori permette anche l'esecuzione simultanea delle funzioni suddette.

I chip di memoria

I chip di memoria sono l'altra importante serie di chip presenti sulla scheda madre. Ci sono due tipi di chip di memoria: RAM e ROM. RAM è l'abbreviazione di Random Access Memory (memoria ad accesso casuale), ROM è acronimo di Read Only Memory (memoria a sola lettura). Entrambe le memorie immagazzinano dati, ma in modo diverso.

La RAM, chiamata anche memoria di lettura e scrittura, è cancellabile, in modo che il microprocessore e i coprocessori possano immettervi dei dati, riprenderli poi in un secondo tempo e cancellarli per fare spazio a nuove informazioni. Ricordatevi che non è possibile conservare dati nella RAM, perché la RAM ha la spiacevole peculiarità di perdere tutti i dati che ha memorizzato quando il computer viene spento.

La ROM, invece, non è cancellabile; i dati immessi nella ROM (chiamati anche "firmware") sono sempre a disposizione dell'Amiga. Il microprocessore e i coprocessori possono leggere i dati della ROM, ma non possono cancellarli, né sostituirli con nuovi dati. La ROM dell'Amiga è occupata dal software sistema (immesso durante la fabbricazione) che il computer usa per poter funzionare non appena viene acceso.

Gli altri chip e componenti elettronici montati sulla scheda madre si occupano dei dettagli concernenti l'invio di dati e informazioni audiovisive dall'unità centrale alle altre parti del sistema. Anche questi componenti sono di importanza fondamentale, ma è innegabile che i chip più importanti della scheda siano il microprocessore, i coprocessori e i chip di memoria.

Il disk drive interno

Il disk drive inserito nel corpo macchina è chiamato disk drive interno, per distinguerlo dai drive che possono essere collegati all'unità via cavo (definiti, per semplicità, disk drive esterni). I disk drive sono necessari per leggere i dati memorizzati su disco e per salvare dati che sarebbero altrimenti cancellati dalla RAM quando il computer viene spento.

Il disk drive interno dell'Amiga richiede l'uso di dischetti da 3,5", che si possono tranquillamente portare in giro nel taschino della giacca. Il drive interno è a doppia faccia, registra cioè dati su entrambi i lati del disco. Il disco ha dunque una capacità due volte superiore a quella che avrebbe se fosse utilizzato con un drive a faccia singola. Il drive dell'Amiga

consente di memorizzare 876 kilobyte di dati su ogni dischetto, una capacità che, se si considerano le ridotte dimensioni del supporto, può essere giudicata sorprendente.

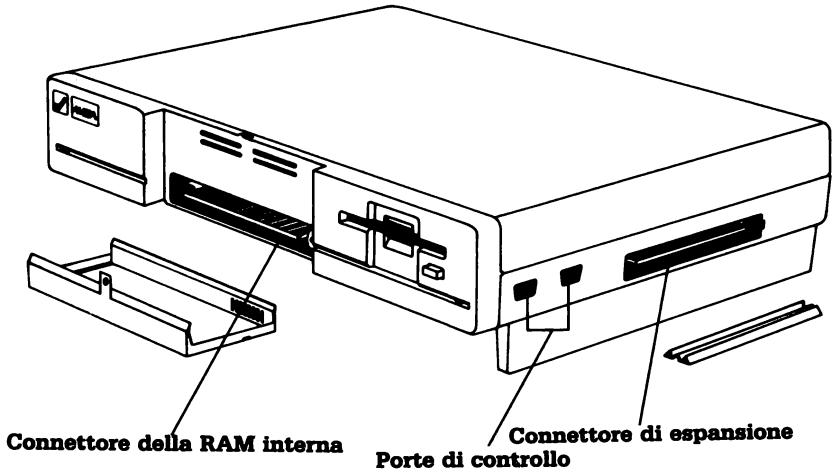
Per immagazzinare e rileggere i dati su disco, l'Amiga utilizza un metodo speciale definito "accesso diretto alla memoria" (DMA – Direct Memory Access). In poche parole, il DMA fa sì che i dati vadano direttamente dal disk drive alla memoria centrale o viceversa, senza passare dal microprocessore. Questo sistema rende molto veloce il trasferimento di dati dal disk drive alla memoria. Inoltre il disk drive può in questo modo trasferire dati da e per la RAM senza interrompere le operazioni in corso. È un vantaggio che si rivela molto utile quando l'Amiga è occupato nell'animazione di figure o nell'emissione di suoni: non deve infatti interrompere le funzioni in corso per ricevere dal disk drive informazioni sulla prosecuzione dell'animazione o sulla nuova canzone da suonare.

PORTE E CONNETTORI

Esternamente al corpo macchina sono collocate le porte per il collegamento delle periferiche. Queste prese servono da ponti per il passaggio di dati dai chip presenti all'interno dell'Amiga alle apparecchiature dislocate nel mondo esterno. Nella Figura 1-5 si possono osservare le porte e i connettori distribuiti sul lato anteriore e destro dell'unità centrale dell'A1000. La Figura 1-6 riporta tutte le uscite presenti sul retro dell'unità centrale.

Figura 1-5.

Le porte e i connettori collocati sui lati frontale e destro dell'unità centrale dell'A1000.



Il connettore della RAM interna

La parte anteriore dell'unità centrale dell'A1000 è dotata di un pannello che cela il connettore della RAM interna. Per

accedere al connettore è sufficiente rimuovere il pannello. Questo connettore viene utilizzato per espandere la memoria dai 256 kilobyte originari dell'A1000 a 512 kilobyte, in modo da aumentarne le possibilità di immagazzinare programmi e dati.

L'A500 è già provvisto di serie di 512 kilobyte di memoria RAM, tuttavia può essere ulteriormente espanso attraverso un'apposita scheda il cui connettore dedicato è collocato nella parte inferiore destra del corpo macchina. La memoria aggiuntiva (512K) che la scheda di espansione (equipaggiata anche di orologio/calendario) mette a disposizione non funziona nello stesso modo della RAM di base, ed è definita "fast RAM". In essa vengono automaticamente memorizzati i programmi caricati da disco, poiché in quest'area vengono eseguiti più velocemente che nella RAM standard.

L'A2000 dispone di 1 MB di memoria RAM che può essere ulteriormente espansa attraverso l'ausilio di apposite schede da collocarsi all'interno del computer.

Le porte di controllo

Sul lato destro dell'Amiga 1000 (anteriormente nell'A2000 e posteriormente nell'A500) sono presenti due porte di controllo. Il mouse dell'Amiga deve essere collegato alla prima di queste porte. Comunque, le porte di controllo non servono solo per il mouse. È possibile inserirvi altri dispositivi di controllo, quali joystick, penne ottiche e tavolette grafiche. Le porte di controllo trasmettono le informazioni provenienti da tali mezzi al chip Paula, che ne legge la posizione o le azioni. Le due porte consentono di usare due sistemi di controllo contemporaneamente.

Gli slot di espansione

Sul lato sinistro del corpo macchina dell'A500 e su quello destro dell'A1000 è presente un altro connettore. Il pannello di plastica che lo ricopre può essere rimosso facilmente con una biro o con le dita. Una volta rimossa la protezione, vedrete un'estremità della scheda madre dotata di una dentellatura in rame, che è appunto il connettore dell'espansione (non toccate assolutamente la dentellatura, una scintilla di elettricità statica delle vostre dita potrebbe danneggiare l'Amiga).

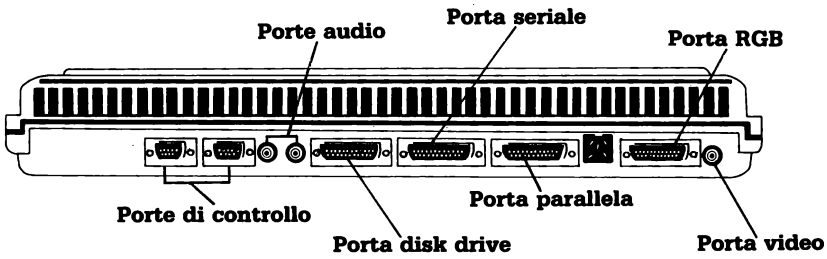
Il connettore per l'espansione consente l'accesso diretto ai chip presenti all'interno dell'unità centrale. Il connettore fa dell'Amiga quello che viene definito un "sistema aperto", cioè un sistema a cui è possibile aggiungere componenti che richiedono necessariamente un contatto diretto con i chip interni. Il connettore per l'espansione permette all'utente di adattare l'Amiga alle sue esigenze.

Componenti collegabili per mezzo di questo connettore sono, ad esempio, un hard disk che aumenti la capacità di memorizzazione dati, chip di RAM aggiuntiva per espandere la memoria (fino a 8 megabyte in più dei 512K della memoria interna di A500 e A1000) e un convertitore del segnale video che consenta l'immissione diretta nell'Amiga di immagini provenienti da fonti esterne, come ad esempio telecamere. La maggior parte di queste espansioni sono dotate di un connettore addizionale compatibile con quello dell'Amiga e consentono dunque all'utente di aggiungere anche più componenti senza "abusare" del connettore Amiga (se rimuovete il pannello di plastica per osservare il connettore, ricordatevi di rimetterlo al suo posto. È una protezione necessaria a evitare che, inavvertitamente, l'utente faccia andare l'Amiga in corto circuito).

L'A2000, a differenza degli altri due modelli della linea Amiga, è espandibile internamente attraverso nove appositi slot destinati ad accogliere schede con funzioni diverse (dalle espansioni di memoria a nuovi processori per la compatibilità con altri sistemi, come ad esempio IBM PC XT ed AT). Un altro connettore, chiamato CPU slot, consente infine l'inserimento di altre MMU e CPU.

Figura 1-6.

Le porte e i connettori sul retro del corpo macchina dell'A500.



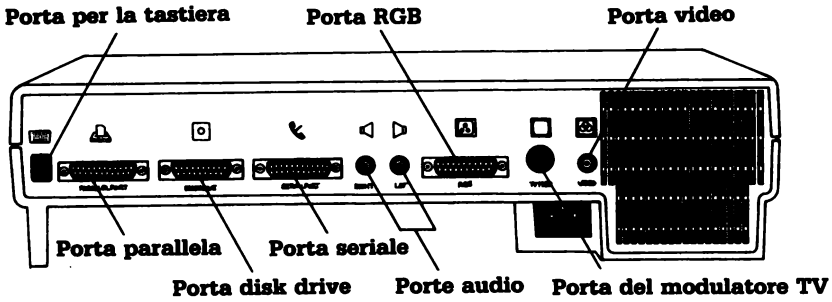


Figura 1-6/b.

Le porte e i connettori sul retro dell'unità centrale dell'A1000.

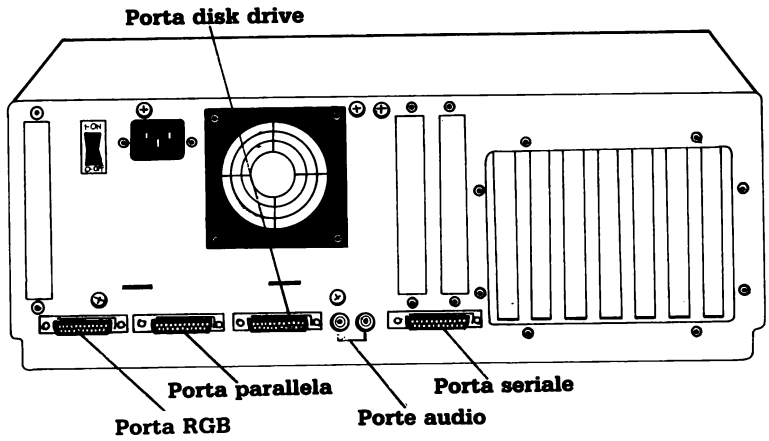


Figura 1-6/c.

Le porte e i connettori sul retro dell'unità centrale dell'A2000.

La porta per la tastiera

Guardando il retro del corpo macchina dell'Amiga, si può osservare tutta la serie di porte illustrate nella Figura 1-6. La prima porta a sinistra è la porta per la tastiera (collocata nella parte frontale nell'A2000 e assente nell'A500, dal momento che è dotato di tastiera integrata), il cui compito è quello di trasmettere informazioni dalla tastiera all'unità centrale. Il cavo della tastiera deve essere connesso a questa presa.

La porta parallela

La seconda presa da sinistra (quarta da destra nell'A500) è la porta parallela. Questa porta invia dati dall'unità centrale a qualsiasi altra periferica sia stata collegata (normalmente una stampante). La maggior parte delle stampanti parallele adottano un cavo e un metodo di trasmissione dati standard, detto "Centronics parallel" (parallelo Centronics), rifacendosi alla serie di stampanti create dalla Centronics Corporation. La porta parallela dell'Amiga 500 e 2000 segue questo standard, mentre quella dell'A1000 utilizza uno standard Centronics modificato; con un cavo apposito è comunque possibile collegare tutte le stampanti predisposte per lo standard Centronics.

La porta disk drive

A destra della porta parallela (sesta da destra nell'A500) è presente la porta per il disk drive, alla quale è possibile collegare un massimo di tre drive (due nell'A2000) per aumentare le capacità di memorizzazione dati del sistema. Per collegare più di un drive a questa porta è sufficiente creare una sorta di catena, connettendo ogni drive aggiuntivo al retro dell'ultimo drive collegato. Tutti i drive accedono direttamente alla RAM (senza passare attraverso il microprocessore), permettendo così un veloce trasferimento dei dati.

Il modello di base A2000 è dotato di un drive interno da 3,5" al quale possono essere aggiunti un altro drive da 3,5" e un ulteriore drive da 3,5" (montato su uno speciale supporto) oppure un drive da 5,25" o un hard disk.

La porta seriale

La porta seriale si trova alla destra (alla sinistra nell'A500) della porta per il disk drive. La porta seriale serve a collegare periferiche alle quali il computer deve inviare necessariamente dati seriali, ad esempio un modem (apparecchio che trasferisce dati dall'Amiga ad un altro computer attraverso collegamento telefonico) o una stampante seriale. Anche per la trasmissione dei dati seriali esiste un sistema standard, denominato "RS232". Questo sistema stabilisce il modo in cui devono essere usati i fili di un cavo di connessione e il modo in cui i bit vengono trasmessi attraverso i fili predisposti. La porta seriale dell'Amiga 500 e del 2000 adotta lo standard RS232, a differenza di quella dell'A1000 che si serve di un RS232 modificato. Con l'ausilio di un cavo apposito è comunque possibile collegare qualsiasi periferica seriale predisposta per lo standard RS232.

Le porte audio

Le due porte audio sono collocate a destra della porta seriale nell'A1000, a sinistra della porta disk drive nell'A500 e a sinistra della porta seriale nell'A2000. La loro funzione è quella di emettere i segnali audio che costituiscono il suono dell'Amiga. È presente una porta audio per il canale sinistro e una per il canale destro.

Per ottenere una buona qualità di riproduzione sonora, le porte audio possono essere collegate a un impianto stereo, nello stesso modo in cui collegheresteste una piastra di registrazione o un lettore di compact disc. L'output delle porte audio è identico a quello di un registratore, quindi il segnale deve essere amplificato. Con l'ausilio di normali cavi audio è possibile collegare le due porte a un amplificatore stereo, sfruttando le prese del registratore, del piatto o di altri componenti. Se volete registrare la musica creata con l'Amiga è sufficiente collegare le porte audio alle prese input di un registratore.

La porta RGB

Nell'Amiga 1000, la porta a destra delle due prese audio è la porta RGB. Nell'A500 è la seconda da destra, mentre nell'A2000 è la prima da sinistra. È qui che dovete inserire il cavo nel caso usiate un monitor RGB. La porta funziona sia con i monitor analogici, sia con quelli digitali, per i quali è però necessario uno speciale cavo adattatore.

La porta del modulatore TV

La porta RGB è seguita, sempre a destra, dalla porta per il modulatore TV (opzionale su A500 e 2000). Questa porta emette un segnale standard videocomposito unitamente ai segnali audio relativi al canale destro e sinistro. Il modulatore per questa uscita è stato annunciato ma non è mai stato realizzato direttamente dalla Commodore. Per utilizzare l'A1000 con un televisore si consiglia di servirsi del modulatore A520 connesso con la porta RGB.

La porta video

La porta video (non presente sull'A2000, monocromatica sull'A500) è quella situata all'estrema destra della parte posteriore del corpo macchina. Questa porta invia un segnale PAL per monitor videocompositi, lo stesso tipo di segnale emesso dal jack VIDEO OUT presente nella maggior parte dei videoregistratori.

Se per il vostro computer utilizzate un monitor videocomposito, dovete inserire il cavo in questa presa. Se usate un monitor diverso (un televisore o un RGB), potete collegare a questa presa un videoregistratore e registrare le immagini visualizzate sullo schermo.

UNITÀ PERIFERICHE

È possibile ampliare il sistema Amiga di base con periferiche che ne aumentino la potenza e rendano l'Amiga stesso più funzionale. Esse consentono un notevole risparmio di tempo e permettono di trasferire i risultati del lavoro dell'Amiga su carta o via cavo telefonico.

Uno dei dispositivi ausiliari più utili all'utente di un Amiga è la stampante. Sono disponibili sul mercato modelli dalle caratteristiche più diverse, comunque la maggior parte di essi è collegabile all'Amiga. Quasi tutte le stampanti vengono usate esclusivamente per stampare testi, ma molte sono anche in grado di riprodurre immagini dallo schermo dell'Amiga. Queste stampanti grafiche sono uno strumento indispensabile agli artisti di computer graphics.

Stampanti a margherita (letter quality)

Nel mondo del lavoro vengono spesso utilizzate stampanti di qualità (letter quality), capaci di scrivere con i caratteri di una vera e propria macchina da scrivere. Queste stampanti imprimono le lettere sulla carta colpendo un nastro inchiostrato con caratteri completamente definiti posti sulla punta di minuscoli bracci, proprio come nelle macchine da scrivere. Benché la loro stampa sia chiara e nitida, queste stampanti non sono molto utili per la grafica, in quanto possono riprodurre solo caratteri e non immagini.

Stampanti a matrice di punti

La stampante a impatto a matrice di punti è quella più usata per i computer. Utilizza una testina di stampa dotata di una fila verticale di minuscoli aghi, che combinandosi con modalità diverse creano i caratteri colpendo un nastro inchiostrato mentre la testina si muove lungo il foglio. Osservando da vicino i caratteri si nota chiaramente che sono formati dai punti stampati dagli aghi. Con gli stessi minuscoli punti la stampante crea anche immagini. La maggior parte delle stampanti a impatto funziona in bianco e nero, ma alcune dispongono di nastri colorati per creare riproduzioni a colori.

Stampanti termiche

Le stampanti termiche funzionano in modo molto simile alle stampanti a impatto; fanno però uso di aghi termici per imprimere direttamente su una carta speciale dei punti simili a quelli creati dalla stampante a impatto. Anche le immagini delle stampanti termiche sono simili a quelle delle stampanti a impatto: alcune dispongono addirittura di un nastro colorato per riprodurre immagini a colori. La grafica delle stampanti termiche in genere non è all'altezza di quella delle stampanti a impatto (i punti non sono disposti con altrettanta cura sulla carta e talvolta non sono ben nitidi), ma è importante sottolineare che le stampanti termiche sono anche molto meno costose.

Stampanti a getto d'inchiostro

Le stampanti a getto d'inchiostro spruzzano piccole gocce d'inchiostro sulla carta per formare lettere e immagini. La maggior parte dei modelli stampa in bianco e nero, ma sono disponibili modelli più costosi che utilizzano una vasta gamma di colori usando getti distinti. Vale la pena di acquistarle se si richiede una grafica di stampa ad alta qualità, perché queste stampanti offrono una colorazione viva e uniforme.

Stampanti laser

La stampante laser è un tipo di stampante che solo pochi si possono permettere. Crea immagini e testi facendo uso di un rullo e di un toner fotosensibili, come una fotocopiatrice, ma crea l'immagine sul rullo con un laser anziché esporre il rullo alla luce riflessa da un foglio attraverso una lente. Le stampanti laser sono molto rapide e silenziose e creano immagini nitide e chiare. Quasi tutte le stampanti laser utilizzano un solo colore. È possibile variare il colore usando apposite cartucce, ma il testo e le immagini di ogni singola pagina risulteranno stampati in un solo colore, a meno che ciascuna pagina non venga inserita più volte nella stampante per essere reimpressa con colori differenti. Grazie al continuo evolversi della tecnologia laser, sarà probabilmente possibile, entro breve tempo, creare stampanti laser che stampino contemporaneamente in molteplici colori sulla stessa pagina. Anche i prezzi dovrebbero calare, permettendo così di allargare lo spettro di utenza ai possessori di micro e personal computer.

Disk drive esterni

È possibile collegare all'Amiga fino a tre (due con l'A2000) floppy disk drive esterni, usando l'apposita porta sul retro dell'unità centrale. Si possono utilizzare drive esterni a doppia faccia da 3,5" (vedi Figura 1-7) o drive speciali IBM compatibili da 5,25". I drive da 5,25" si servono di dischi di dimensioni maggiori che oltretutto consentono di memorizzare una quantità di dati inferiore ai normali dischi Amiga da 3,5"; tuttavia queste periferiche sono in grado di leggere dischi PC IBM e quindi consentono l'uso del software IBM o la lettura di dati creati da un PC IBM.

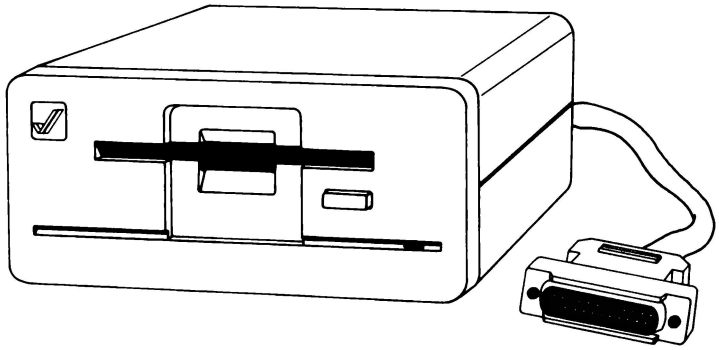


Figura 1-7.

Un disk drive esterno Amiga da 3,5".

Ogni disk drive da 3,5" fornisce 876 kilobyte di memoria su disco, sufficienti già di per sé ad immagazzinare l'intero contenuto di questo libro. Il disk drive addizionale facilita

anche la copiatura di file da un dischetto all'altro. Infatti, usando solo il disk drive interno, l'utente, per copiare file da un dischetto all'altro, deve continuamente inserire e togliere i supporti magnetici dal drive. Un secondo disk drive elimina il continuo avvicendamento dei dischetti, assicurando un notevole risparmio di tempo.

Per ottenere una maggiore capacità di registrazione di dati è possibile servirsi di un hard disk, apparecchio in grado di immagazzinare diversi megabyte (cioè milioni di byte) di informazioni. Questo considerevole spazio di memoria è generalmente utilizzato per la gestione dei record di un database, ma si può rivelare utile anche per creatori di animazioni e musicisti che desiderino immagazzinare complesse sequenze animate o spartiti molto lunghi.

L'uso di un hard disk drive presenta anche un altro vantaggio: consente infatti di trasferire informazioni da e per l'Amiga molto più velocemente di un floppy disk drive. Tale velocità operativa può rivelarsi utile qualora, ad esempio, si intenda inserire in memoria rapidamente lunghe sequenze di immagini animate.

RAM esterna

L'Amiga 1000 dispone di un massimo di 512 kilobyte di RAM interna. Per espandere la memoria dell'Amiga è possibile aggiungere fino a otto megabyte di RAM esterna, inserendo cartucce RAM nel connettore situato nel lato destro dell'unità centrale. Dal momento che immagini e suoni richiedono grandi quantità di memoria, l'extra RAM può essere utile per immagazzinare complesse sequenze animate o lunghe partiture musicali.

L'Amiga 500 può essere espanso in modo analogo al 1000 utilizzando il connettore situato nel lato sinistro del corpo macchina. L'Amiga 2000, proprio per la sua progettuale "architettura a schede" può essere espanso solo internamente.

Periferiche addizionali

Esiste tutta una gamma di altre periferiche molto utili, ciascuna con compiti specifici. Il modem permette di collegare l'Amiga alla linea telefonica in modo da comunicare con altri computer o reti di terminali. Le tavolette grafiche e le penne ottiche permettono di disegnare figure sul monitor senza usare il mouse. I sintetizzatori possono accrescere ulteriormente le capacità musicali, mentre videocamere, videoregistratori ed altri accessori video consentono lo scambio delle immagini con il mondo esterno. Nei capitoli successivi troverete ulteriori informazioni su queste periferiche addizionali.

IL SOFTWARE

Il software esiste solo sotto forma di stati elettrici nella memoria dell'Amiga o di cariche magnetiche su un dischetto, e pertanto non è facilmente analizzabile come l'hardware. Per esaminarne la struttura è necessario dare un'occhiata ai tabulati del codice usato dal programmatore per creare il software. Ma si tratta di un compito arduo e per nulla divertente (le mie scuse agli hacker più accaniti). Di fatto, il software è ciò che l'Amiga fa, e pertanto il modo più piacevole per esaminarlo consiste proprio nel considerarne i risultati.

IL SOFTWARE SISTEMA

Il software inserito nell'Amiga, detto software sistema, è quello che fa funzionare la macchina. Parte di questo software è contenuta all'interno dell'unità centrale sui chip ROM; il resto è su due floppy disk (uno solo per A500 e 2000): il Workbench e il Kickstart (già implementato su ROM nell'A500 e 2000). Il software sistema dispone di cinque programmi principali e di numerose routine minori, strutturate a piramide in modo da poter lavorare contemporaneamente.

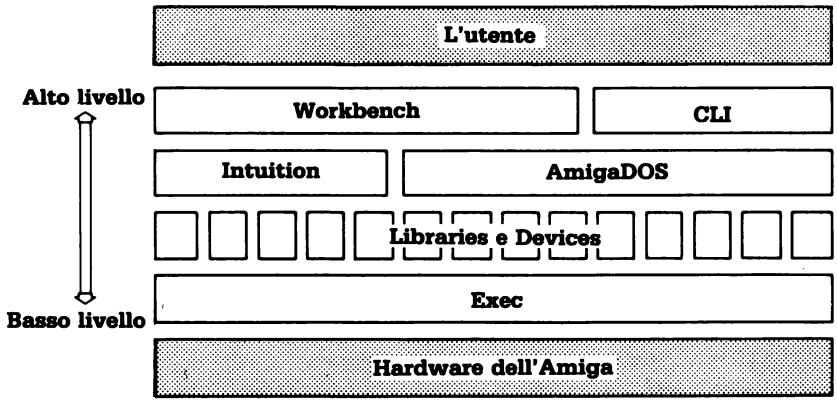
Un paragone permette forse una migliore comprensione del funzionamento del sistema. Si pensi a un'impresa che produce pifferi. Alcuni macchinari tranciano il metallo, altri lo modellano, altri ancora attaccano le membrane vibratili, dipingono il metallo e confezionano il prodotto finito. Le macchine sono azionate da un gruppo di operai, uomini e donne esperti che assicurano il perfetto funzionamento dei macchinari loro affidati. Gli operai sono guidati da supervisori che si assicurano che tutto proceda nel migliore dei modi. I supervisori, a loro volta, sono controllati da manager competenti nei vari settori della produzione di pifferi: un manager per il taglio del metallo, un altro per una colorazione attraente, e così via. Un gradino più in alto vi sono i direttori, uno per ogni area, ad esempio spedizioni o controllo di qualità. I vicepresidenti sono al livello immediatamente superiore ed eseguono gli ordini del presidente dell'impresa costruttrice di pifferi.

Quando il presidente decide di fare qualcosa di nuovo, come introdurre una nuova linea di pifferi, ordina a uno dei suoi vicepresidenti di avviare la nuova produzione. Il vicepresidente dà disposizioni ai direttori che, a loro volta, emanano i loro ordini ai manager. I manager spiegano poi agli operai quali siano i cambiamenti da attuare nell'uso dei macchinari.

Pur non avendo nulla a che fare con i pifferi, si può dire che l'Amiga funzioni secondo gli stessi principi di questa ipotetica impresa. L'hardware (gli operai della fabbrica) svolge il grosso del lavoro, mentre il software (i dirigenti) si limita a dirigerlo. Al livello inferiore del software c'è un programma denominato Exec (i supervisori), che dirige direttamente l'hardware. A dare ordini all'Exec sono alcuni sottoprogrammi detti "libraries" e "devices" (librerie e congegni, i manager), specializzati in settori come grafica, riproduzione sonora e matematica. Al livello immediatamente superiore a libraries e devices vi sono l'AmigaDOS e Intuition (i direttori), due programmi che coordinano il lavoro del software dei livelli inferiori, allo scopo di svolgere determinati compiti come il trasferimento di dati dal disk drive alla scheda madre e viceversa, e la presentazione dei dati in una forma comprensibile all'utente. Al livello superiore del software sistema vi sono altri due programmi, Workbench e CLI (i vicepresidenti) che prendono ordini direttamente dall'utente (il presidente) e li trasmettono al resto del software del sistema. La Figura 1-8 illustra i diversi livelli del software sistema dell'Amiga.

Figura 1-8.

I livelli del software sistema compresi fra l'utente e l'hardware.



La stratificazione del software facilita la creazione di programmi per l'Amiga. I programmatori possono utilizzare i programmi situati ai livelli più bassi del software sistema per semplificare gran parte del proprio lavoro. I programmi ai livelli inferiori, come libraries e devices, dispongono di routine che svolgono compiti generalmente richiesti da tutti i programmi: richiamare dati dal disk drive o leggere quanto digitato dall'operatore sulla tastiera. Il programmatore deve semplicemente chiedere a libraries e devices di svolgere la funzione desiderata, anziché inserire ogni volta nuovi comandi per fare la stessa cosa. Ad esempio, un programmatore che sta scrivendo un nuovo programma non deve preoccuparsi di leggere le

azioni del mouse. Il nuovo programma può invece chiedere a una routine di livello inferiore di leggere le azioni del mouse e inviargli le relative informazioni. Il tutto comporta un risparmio di tempo e fatica per il programmatore.

Exec

Exec (abbreviazione di "system executive") è il programma situato al livello più basso di tutto il software sistema dell'Amiga. È memorizzato nella ROM all'interno dell'unità centrale e nell'Amiga 1000 sul disco Kickstart. Quando si accende il computer, è l'Exec che avvia il sistema. Per prima cosa libera la memoria dell'Amiga, controlla velocemente l'hardware e poi, nell'Amiga 1000, chiede all'operatore di inserire il Kickstart nel disk drive per immettere in memoria anche il resto dell'Exec. Caricato il Kickstart, l'Exec richiede il Workbench per completare l'inserimento in memoria del software sistema.

L'Exec si occupa delle attività base dell'hardware. Dispone di routine che trasmettono i dati da un chip all'altro e li trasferiscono all'esterno attraverso le porte e i connettori. L'Exec dispone anche di routine che controllano la tastiera, seguono le azioni del mouse, generano segnali video e audio e svolgono altri compiti fondamentali per l'operatività dell'Amiga.

L'Exec può svolgere contemporaneamente molti compiti diversi. Ad esempio può trasmettere dati a una stampante mentre muove alcune figure sullo schermo. Questa capacità è definita "multitasking" (polifunzionalità) ed è una delle caratteristiche che fanno dell'Amiga una macchina eccezionale.

Per destreggiarsi con successo nella gestione contemporanea di diverse attività, l'Exec deve assicurarsi che ognuna di esse utilizzi solo i chip che le competono e non interferisca con le altre. L'Exec è un po' come un avvocato al servizio di clienti con interessi contrastanti: dedica a ognuno tanto tempo quanto basta per non farli sentire trascurati e tiene tutti all'oscuro di quanto succede agli altri.

Libraries e devices

Le libraries (o librerie) sono piccoli programmi di livello immediatamente superiore all'Exec. Ogni libreria ha la sua specialità, e messe tutte insieme coprono una vasta gamma di funzioni molto utili: grafica, animazione, calcoli matematici e visualizzazione di testi sullo schermo. Tutte le libraries sono al servizio del software di livello superiore. Ad esempio, un programma che crea immagini video può utilizzare routine proprie della libreria grafica, mentre un programma per fare musica può ricorrere alle librerie audio, consentendo al programmatore di risparmiare tempo e programmazioni ripetitive.

I devices (o congegni) dell'Amiga non fanno parte del-

l'hardware, come invece suggerirebbe il loro nome. Sono anch'essi parte integrante del software, come le libraries, pur essendo generalmente destinati al controllo di una parte specifica dell'hardware interno dell'Amiga, a differenza delle libraries, che adempiono a funzioni meno specifiche. I devices interpretano le pressioni esercitate dall'utente sui tasti, seguono i movimenti e gli scatti del bottone del mouse, trasmettono dati attraverso le porte seriale e parallela e svolgono altre funzioni relative all'hardware. I devices, inoltre, generano i suoni dell'Amiga e tengono conto del trascorrere del tempo.

I programmatori possono ricorrere direttamente all'aiuto dei devices nei loro programmi. Il BASIC Microsoft per l'Amiga, ad esempio, offre comandi speciali che si rivolgono direttamente a libraries e devices per svolgere alcune funzioni che il BASIC da solo non sarebbe in grado di espletare.

AmigaDOS

L'AmigaDOS (abbreviazione per Amiga Disk Operating System) si trova un gradino più in alto di libraries e devices. L'AmigaDOS registra la presenza dei dati e dei programmi conservati nella memoria dell'Amiga o sui dischetti presenti nei drive. Questi programmi e gruppi logici di dati sono detti file. Quando l'AmigaDOS immagazzina un file su un disco, registra la sua locazione, le sue dimensioni e l'ora della memorizzazione. In questo modo, quando si tratta di riportare il file nella RAM, l'AmigaDOS sa benissimo dove trovarlo e come riportarlo nell'unità centrale.

Altra funzione dell'AmigaDOS è l'avviamento di diversi programmi. Dal momento che l'Exec può gestire contemporaneamente più attività, l'AmigaDOS sfrutta la polifunzionalità dell'Exec per far girare più programmi simultaneamente. In caso di conflittualità tra i vari programmi, l'AmigaDOS sa riconoscere quale programma sia più importante e si assicura che esso abbia la precedenza sia nell'elaborazione svolta dai chip, sia nella registrazione dei dati.

L'AmigaDOS viene utilizzato ogni volta che l'utente deve caricare un programma da disco e farlo girare. Per dare ordini all'AmigaDOS è necessario utilizzare uno dei due programmi situati al livello superiore: il Workbench o il CLI.

Intuition

Il programma Intuition opera allo stesso livello dell'AmigaDOS e dispone di routine che ne fanno una sorta di interfaccia fra uomo e macchina. L'interfaccia grafica Intuition è il mezzo usato dall'utente per controllare un programma. Le routine di Intuition creano tutti gli elementi necessari all'interfaccia utente dell'Amiga: schermi, finestre, menu e gadget. Intuition dispone inoltre di routine che ricorrono a libraries e devices

per visualizzare testi ed immagini sullo schermo e che "leggono" il mouse e la tastiera per comprendere ciò che l'utente sta richiedendo.

I programmatori possono utilizzare le routine di Intuition per creare un'interfaccia grafica per i loro programmi. Invece di dover creare menu, finestre e cose simili, il programmatore può delegare quest'onere a Intuition (ad esempio, nel quarto capitolo imparerete a creare una finestra in un programma BASIC). Le caratteristiche di Intuition stimolano tutti i programmatori a servirsi dell'interfaccia grafica in misura consistente, in modo che chi acquista i programmi non sia costretto a dover imparare nuovi comandi a ogni nuova pubblicazione di software.

Workbench

Il Workbench è un programma basato sulla grafica; rappresenta il livello più alto del software sistema dell'Amiga e prende ordini direttamente dall'operatore. Serve da interprete fra l'AmigaDOS e l'utente, consentendo a quest'ultimo di utilizzare l'AmigaDOS per gestire i file presenti su disco o nella RAM. Lo schermo riportato nella Figura 1-9 è un esempio delle visualizzazioni del Workbench.

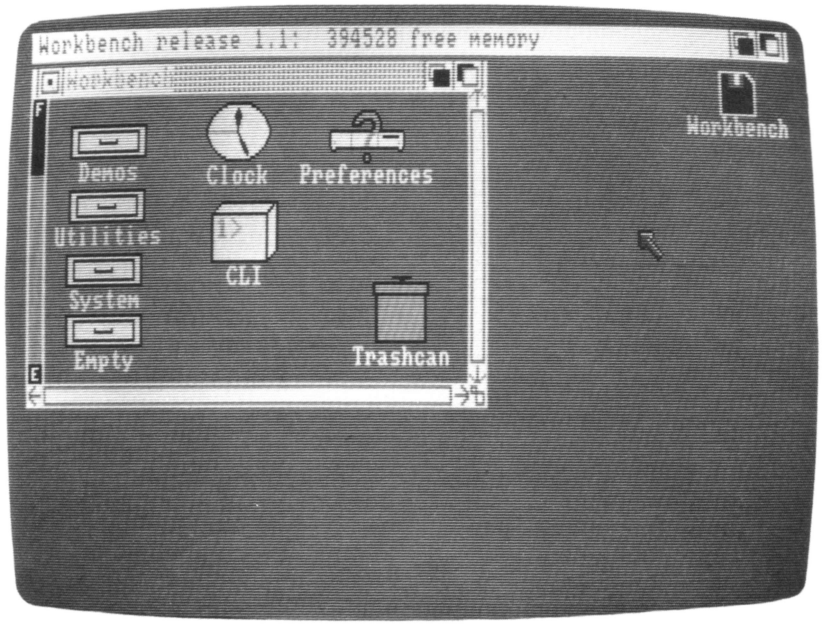


Figura 1-9.

Lo schermo del Workbench.

Il Workbench interpreta le azioni dell'utente sfruttando le routine di Intuition. Quando premete due volte il bottone del mouse per far apparire l'icona di un disco, il Workbench ricorre a Intuition per aprire sullo schermo una finestra entro la quale visualizzare le icone corrispondenti ai file memorizzati sul disco. Il Workbench chiede all'AmigaDOS di controllare quali file contenga il dischetto. Funzioni tipiche del Workbench sono anche la copiatura dei dischetti, l'avviamento dei programmi, la copiatura dei file e la cancellatura dei file da disco.

CLI

CLI, acronimo di Command Line Interpreter (o Interface), è un programma basato sul testo, presente in alternativa alla grafica del Workbench. A differenza di quest'ultimo, il CLI non fa uso delle routine dell'interfaccia Intuition per rendere comprensibili all'AmigaDOS i comandi dell'utente. Al contrario, il CLI adotta un'interfaccia utente meno sofisticata, che richiede la digitazione delle istruzioni in un'apposita linea di comando, il tutto a opera dell'utente. Il CLI interpreta ogni comando come un'attività dell'AmigaDOS e avvia la funzione richiesta dall'operatore. La Figura 1-10 illustra un esempio di istruzioni AmigaDOS fornite utilizzando il CLI.

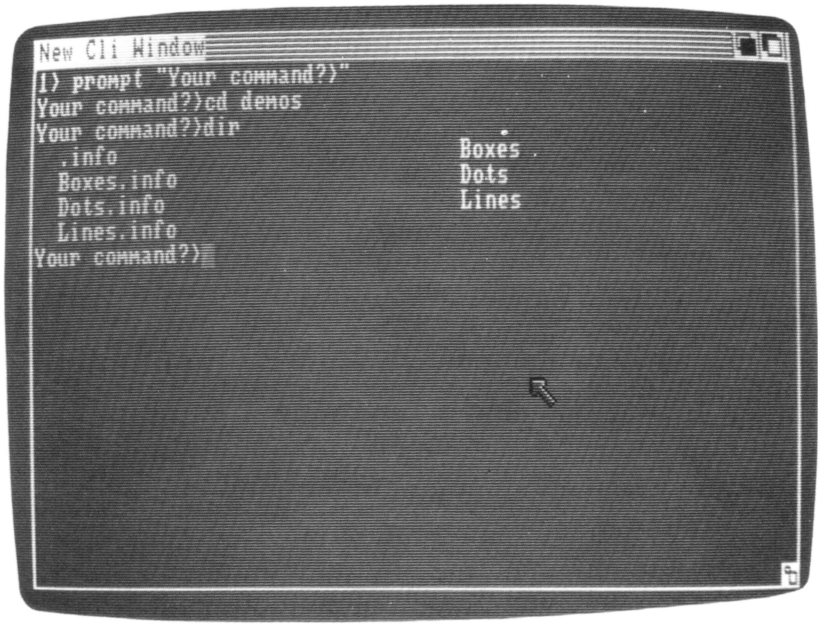


Figura 1-10.

L'AmigaDOS utilizzato con i comandi CLI.

Sebbene non risulti sempre di facile uso, il CLI è senz'altro più potente del Workbench e permette di usufruire di un maggior numero di funzioni dell'AmigaDOS. Esempi di attività del CLI non riscontrabili nel Workbench sono l'avviamento di un programma non appena viene acceso il computer e la ricerca di un dato particolare all'interno di un gruppo di file. È inoltre possibile servirsi del CLI per eseguire una serie di comandi AmigaDOS e per creare programmi AmigaDOS che si occupino della gestione di file, specialmente se lunga e complessa.

Il manuale fornito con il computer non dà alcuna indicazione su come sfruttare il CLI. Potete trovare tutte le informazioni necessarie nel *Manuale dell'AmigaDOS* (pubblicato dalla IHT Gruppo Editoriale) in vendita nelle migliori librerie.

STRUMENTI SOFTWARE

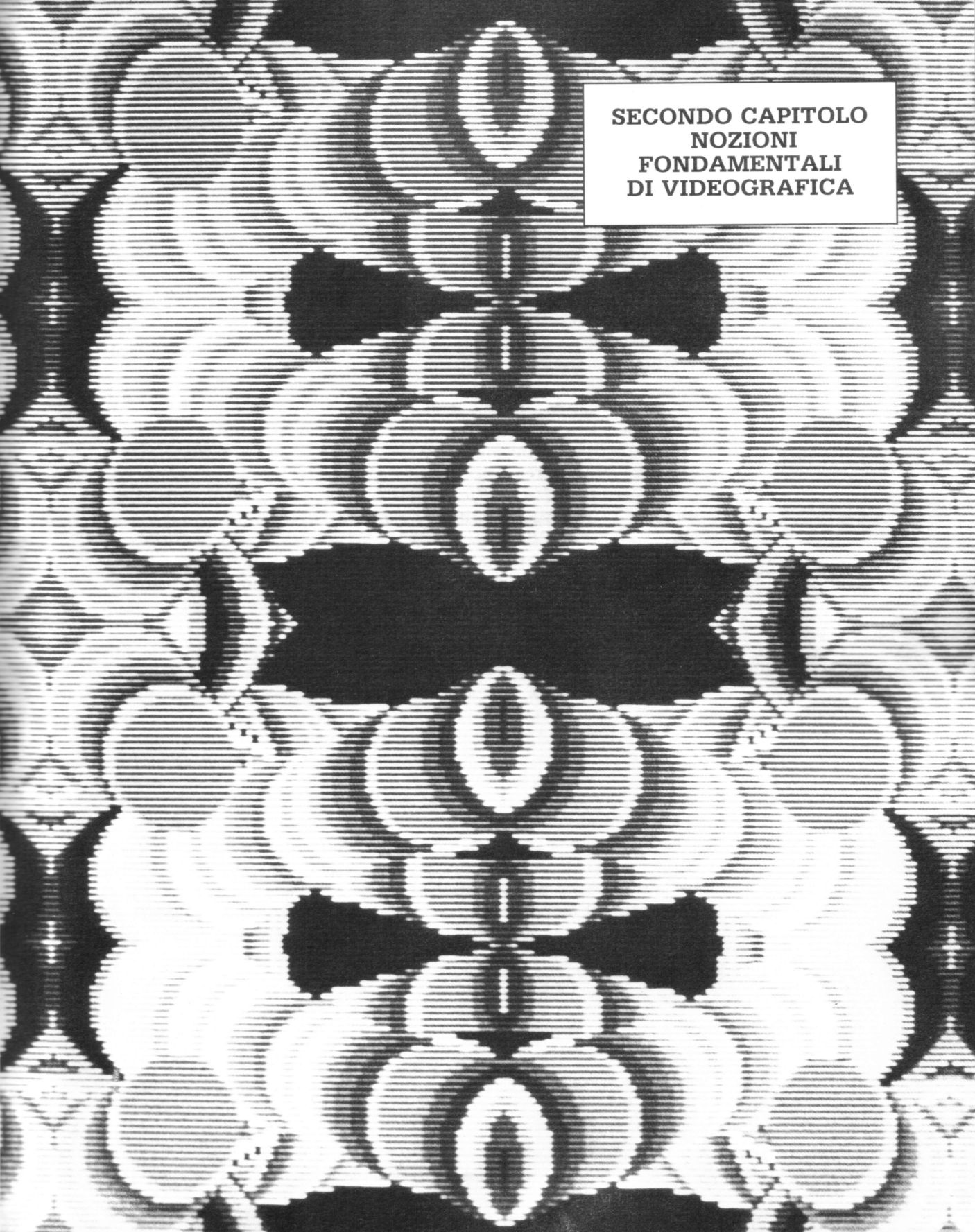
Il software sistema non è che una minima parte del software che potete usare con l'Amiga. Per ottenere risultati tangibili, vi suggeriamo di provare programmi come Deluxe Paint, Deluxe Music, Deluxe Video e l'Amiga BASIC, descritti nelle altre sezioni di questo volume. Questi programmi si rivolgono ad aree di utilizzo specifiche, facendo il miglior uso del software sistema allo scopo di consentire all'utente di ottenere i risultati desiderati.

Si conclude qui la descrizione del sistema Amiga, interno ed esterno. Le diverse componenti, hardware e software, collaborano per fare dell'Amiga una macchina davvero sbalorditiva. Ma un computer, per quanto potente, resta pur sempre uno strumento, e a lungo andare è l'utente che, prendendo confidenza con la macchina e scoprendone pian piano le enormi capacità, svolge il lavoro vero e proprio, creando le immagini, i suoni e le animazioni che rivelano tutto il "talento" dell'Amiga.

SEZIONE 2

Immagini

Questa sezione è dedicata alla creazione di immagini con il computer Amiga. Il secondo capitolo illustra le nozioni basilari della video grafica, mentre il terzo capitolo è dedicato all'uso del programma applicativo di grafica Deluxe Paint. Seguono brevi cenni ad altri programmi grafici per l'Amiga e a una serie di periferiche collegabili alla macchina. Il quarto, quinto e sesto capitolo sono dedicati alla trattazione dei comandi grafici dell'Amiga BASIC. Al termine di questa sezione avrete acquisito le basi necessarie a fare di voi un vero artista dell'Amiga.



SECONDO CAPITOLO NOZIONI FONDAMENTALI DI VIDEOGRAFICA

Ognuno di noi vede ogni giorno esempi di videografica sotto forma di spot televisivi. Tuttavia, nonostante la loro onnipresenza, solo pochi sanno come vengano creati. A differenza di mezzi espressivi più semplici, come gli acquerelli o la penna e l'inchiostro, il video utilizza un'apparecchiatura complessa per produrre le immagini. In ogni caso, come per qualsiasi altro strumento, la qualità delle immagini video è determinata dalle proprietà dei suoi elementi costitutivi. Così come il tipo di carta e la trasparenza dei colori degli acquerelli caratterizzano chiaramente la pittura, i fosfori luminosi sullo schermo del monitor conferiscono alle immagini video proprietà non riscontrabili in nessun altro mezzo di comunicazione visiva.

Per creare le migliori immagini sul monitor, è necessario conoscere la complessità del video, oltre a ciò che l'Amiga è in grado di fare attraverso le routine di elaborazione grafica di cui è dotato il suo software sistema. Nei capitoli successivi, apprenderete come applicare direttamente le capacità grafiche dell'Amiga, disegnando e muovendo figure sullo schermo. Prima di tutto è però necessario conoscere le due componenti principali che determinano il modo in cui percepiamo le immagini video: il monitor che visualizza l'immagine e l'occhio che la riceve.

LA PERCEZIONE DELLA LUMINOSITÀ E DEI COLORI

L'occhio umano è un organo complesso. Percepisce forme, distanze, luminosità e colori grazie alla luce che passa attraverso le pupille situate nella parte anteriore dell'occhio. Sebbene percepire le forme e valutare le distanze sia evidentemente di grande rilevanza per la vita di tutti i giorni, per l'artista gli elementi più importanti sono spesso la luminosità e il colore. Per le immagini video luminosità e colore sono ancora più rilevanti, e il monitor dell'Amiga è stato ideato appositamente per creare immagini che sfruttino al meglio il modo in cui l'occhio umano percepisce i colori e la luminosità.

I COLORI FONDAMENTALI

Quando si guarda un'immagine su un monitor a colori, i colori che si vedono sono, in realtà, illusori. In effetti, il monitor può visualizzare solo tre colori fondamentali: il rosso, il verde e il blu. L'Amiga combina e associa questi colori fondamentali e, dalle combinazioni risultanti, crea tutti quei colori che crediamo di vedere. Questo effetto è direttamente riconducibile al modo in cui l'occhio percepisce i colori.

Nella parte posteriore del bulbo oculare esistono due tipi di recettori luminosi: i bastoncelli e i coni. Mentre i coni percepiscono i colori, i bastoncelli vedono un mondo privo di tinte, con sole sfumature di grigio. Sebbene si percepisca un'intera gamma di colori, rosso, giallo, verde, azzurro, blu intenso, viola e molti altri, i coni dell'occhio possono in realtà vedere solo i tre colori fondamentali. Il cervello, unendo i diversi fasci di luce verde, rossa e blu che colpiscono l'occhio, crea nuovi colori.

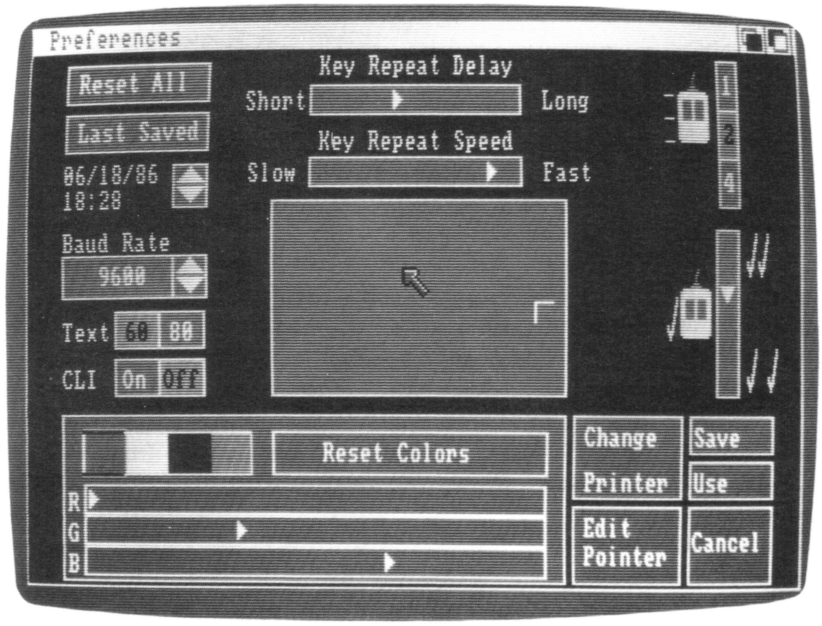
I colori che ne derivano sono determinati dalla proporzione dei colori fondamentali associati. Il giallo, ad esempio, risulta dalla combinazione di quantità uguali di luce rossa e verde. Se la luce rossa è presente in quantità maggiore rispetto a quella verde, il giallo assume sfumature di arancione. Se invece è predominante la luce verde, il giallo diventa verdognolo. Variando la quantità di luce rossa, verde e blu che entra nell'occhio, è possibile creare un'ampia gamma di colori.

LA CREAZIONE DEI COLORI RGB

Per creare colori differenti utilizzando i programmi grafici dell'Amiga, si possono regolare i parametri di controllo dei colori del programma stesso in modo da variare le quantità di rosso, verde e blu, combinate per ottenere un altro colore. Si tratta del cosiddetto metodo RGB per la creazione dei colori. Se ad esempio l'utente seleziona lo schermo Preferences del Workbench, gli vengono mostrate tre regolazioni tramite cursori. Con il metodo RGB si ottengono nuovi colori regolando i tre diversi cursori che controllano la percentuale di ogni colore fondamentale. Regolando i cursori (vedere Figura 2-1 nella pagina successiva) su diverse quantità di colore, viene a modificarsi la proporzione di rosso, verde e blu, e si dà origine a un colore del tutto nuovo.

Figura 2-1.

I controlli RGB del programma Preferences. Muovendo i cursori verso sinistra, l'intensità dei colori fondamentali diminuisce; muovendoli verso destra, invece, aumenta.



LA CREAZIONE DEI COLORI HIS

Sebbene non comporti difficoltà stabilire le combinazioni di rosso, verde e blu per la creazione di nuovi colori, non è sempre semplice descrivere i colori in termini di percentuali di rosso, verde e blu o ottenere l'esatto colore che ci si era prefissati. È d'uso comune definire i colori utilizzando termini quali "color porpora chiaro" o "rosso scuro", piuttosto che descrizioni del tipo "otto parti di rosso, otto di blu e tre di verde". Un altro metodo per caratterizzare i colori si serve di tre caratteri distintivi: tonalità, intensità e saturazione (hue, intensity and saturation, da cui HIS). Questi caratteri descrivono i colori più verosimilmente rispetto alla nostra percezione.

La tonalità

La tonalità è il carattere più evidente del colore. Di solito si fa riferimento alla tonalità, quando l'argomento in questione è il colore, utilizzando termini quali rosso, porpora, arancione e verde. La fusione dei colori fondamentali in un unico colore definisce la tonalità, che cambia a seconda della proporzione dei colori base. Ad esempio, una combinazione di luce blu e verde, in cui l'intensità della luce blu è doppia rispetto a quella della luce verde, dà origine a una tonalità blu-verdognola. Una combinazione, invece, di luce blu e verde in cui è l'intensità della luce verde a essere doppia rispetto a quella della luce blu, crea una tonalità che potremmo definire verde-bluastra. Essendo spesso la tonalità il risultato dell'associazio-

ne di più tonalità diverse, molte vengono indicate con nomi come arancione rossiccio, rosa violaceo e marrone giallognolo.

L'intensità

L'intensità complessiva della luce determina l'intensità del colore. L'intensità può variare dallo scuro (in assenza completa di luce) allo splendente (intensità massima di luce). Come esempio di diversa intensità di colore basti pensare alla differenza esistente fra i colori alle prime luci dell'alba e i colori quando il sole è alto nel cielo. Molte tonalità di colore sono identiche, ma la loro intensità aumenta con l'avanzare del giorno e pertanto i colori risultano sempre più luminosi.

La tonalità e l'intensità sono due fattori ben distinti: l'una può cambiare senza alterare l'altra. Ad esempio, una tonalità blu-verdognola risulta dall'unione di due parti di luce blu e una parte di luce verde, con una proporzione dunque di due a uno. Se l'intensità di entrambe le luci viene raddoppiata, la nuova fusione risulta formata da quattro parti di luce blu e due parti di luce verde; la proporzione resta di due a uno, quindi la tonalità è ancora blu-verdognola. L'intensità però cambia e la seconda combinazione di colori risulta di gran lunga più luminosa.

Saturazione

La saturazione di un colore cambia sulla base della quantità di bianco a esso miscelata. I colori completamente saturi non presentano alcuna traccia di bianco e quindi appaiono più vividi. I colori, invece, poco saturi, perché diluiti col bianco, sono tenui come i colori pastello.

Essendo il bianco luminoso semplicemente il risultato di una combinazione di uguali quantità di rosso, di verde e di blu, quando si aggiunge bianco a un colore, in realtà si aggiungono rosso, verde e blu in parti eguali. Pertanto, la saturazione è legata alla tonalità: entrambe subiscono dei cambiamenti in seguito all'aggiunta di bianco, poiché rosso, verde e blu in parti uguali, utilizzati per creare il bianco, modificano la proporzione dei colori fondamentali all'interno del colore ottenuto. Stabilire se un colore muti tonalità o saturazione è essenzialmente una questione di carattere soggettivo (il nuovo colore appare più tenue o si presenta veramente come una nuova tonalità?).

Alcuni programmi grafici dell'Amiga utilizzano l'HIS per creare nuovi colori. Ad esempio, Deluxe Paint, descritto nel prossimo capitolo, si serve di tre cursori per controllare la tonalità, l'intensità e la saturazione di un colore, così come il programma Preferences utilizza tre cursori RGB. Deluxe Paint trasforma le percentuali HIS in percentuali RGB, necessarie all'Amiga per realizzare i colori sullo schermo del monitor.

VISUALIZZAZIONE DELLE IMMAGINI SU MONITOR

Il funzionamento del monitor di un computer o di un televisore è, per molti, non meno misterioso dei processi biochimici dell'occhio umano. I monitor, invece, non dovrebbero essere tanto misteriosi, dal momento che utilizzano principi dell'elettronica noti ormai da tempo.

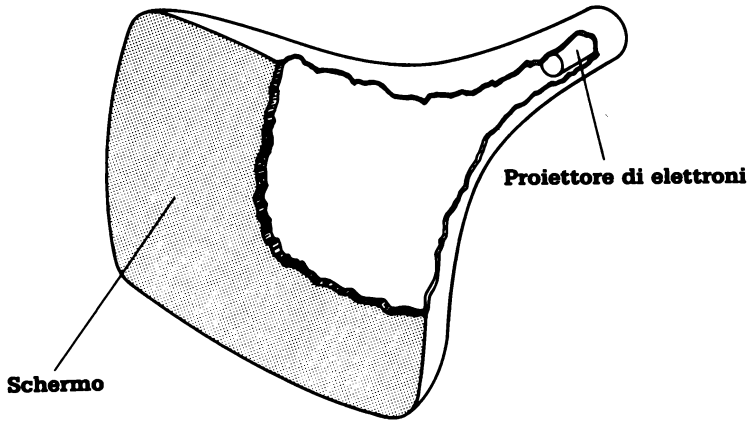
Per imparare a conoscere i monitor, è certamente più utile ai fini didattici un primo approccio con i monitor monocromatici, che visualizzano le immagini utilizzando un unico colore. Nonostante gli utenti di monitor monocromatici collegati all'Amiga siano una minoranza, capire i principi di funzionamento di questi apparecchi video facilita notevolmente l'apprendimento di quelli relativi ai monitor a colori.

I MONITOR MONOCROMATICI

Il cuore di un monitor monocromatico, come di quasi ogni tipo di monitor, è costituito dal tubo catodico (CRT, Cathode-Ray Tube). Il CRT è un largo tubo sottovuoto costituito da uno schermo rettangolare per la visualizzazione delle immagini, dal quale si estende uno stelo allungato contenente un proiettore di elettroni che sporge dalla parte terminale (vedere Figura 2-2).

Figura 2-2.

Sezione di un tubo catodico.



Quando si accende il monitor e si attende che il CRT si scaldi, il proiettore invia un flusso di elettroni alla parte anteriore del tubo catodico, zona coperta da uno strato di particelle di fosforo. Ciascuna particella di fosforo, quando viene colpita da un elettrone, brilla per breve tempo, mentre se non viene colpita resta scura. Nel televisore in bianco e nero, che è il più comune monitor monocromatico, il fosforo, brillan-

do, emana luce bianca. Alcuni monitor monocromatici per computer utilizzano, invece di quelli bianchi, fosfori di luce verde o ambrata.

Il proiettore di elettroni

Per creare un'immagine sullo schermo, il CRT fa convergere il flusso di elettroni inviati dal proiettore in uno stretto fascio luminoso, e dirige tale fascio sulle particelle di fosforo. La Figura 2-3 mostra tale procedimento. Inizialmente il CRT usa campi magnetici per accelerare la velocità degli elettroni quel tanto che basta per far brillare le particelle di fosforo quando sono colpite e, in seguito, sfrutta altri campi magnetici per produrre, attraverso la convergenza degli elettroni, uno stretto fascio luminoso tanto preciso da colpire un solo punto alla volta sull'intero strato di fosforo. Infine, i campi magnetici nell'area di deflessione alzano o abbassano e muovono a sinistra o a destra il fascio di elettroni, dirigendolo su punti diversi dello schermo.

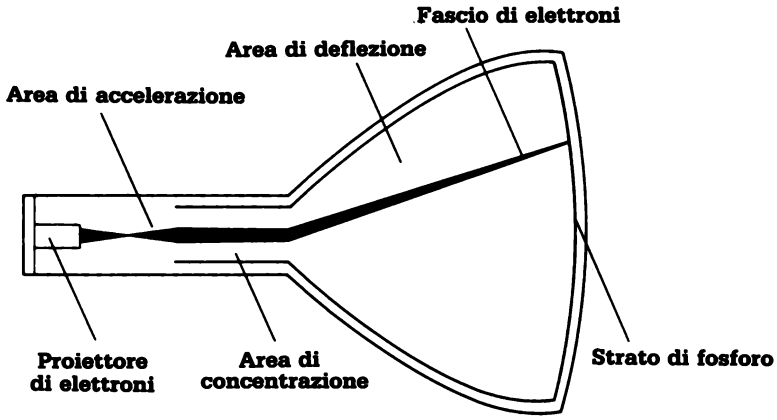


Figura 2-3.

Concentrazione e irradiazione del fascio di un proiettore di elettroni in un CRT.

Scansione del quadro

Per produrre un'immagine su tutto lo schermo, la maggior parte dei monitor utilizza una tecnica denominata scansione del quadro (scanning). L'area dello schermo che viene colpita dal fascio di elettroni è detta quadro. I circuiti elettrici del computer o del televisore che controllano il monitor dividono il quadro in diverse centinaia di linee orizzontali, dette linee del quadro. Inoltre inviano segnali al monitor, che sposta il fascio del proiettore di elettroni sullo schermo per disegnare ogni linea del quadro.

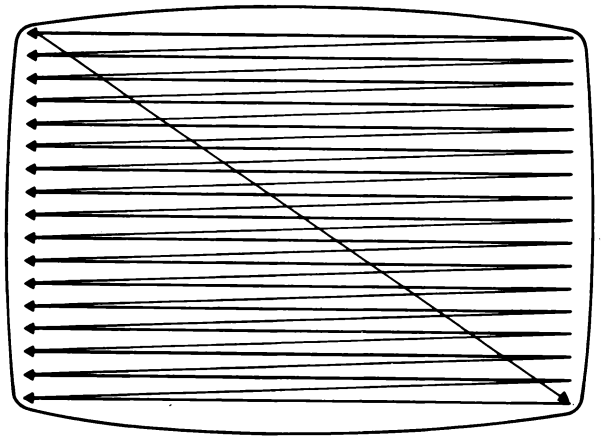
Per coprire l'intero schermo, il fascio di elettroni parte dall'angolo sinistro superiore dello schermo ed "esplora", muovendosi da sinistra verso destra, tutta la parte superiore dello

schermo. L'intensità del fascio di elettroni determina il grado di lucentezza di ciascuna particella di fosforo quando il fascio la colpisce. Quando i fosfori devono rimanere scuri, non si ha alcun invio di elettroni, che si verifica invece quando le particelle devono risultare completamente brillanti. Nel caso di luminosità intermedia, il proiettore invia gli elettroni con minor forza, cosicché la luminosità del fosforo risulta attenuata.

Quando raggiunge l'estremità destra della linea del quadro, il fascio di elettroni ritorna sul lato sinistro dello schermo con un movimento noto come ritorno orizzontale del raggio catodico (horizontal retrace), per poi ricominciare a disegnare una nuova linea appena sotto l'ultima tracciata. Dopo aver tracciato anche questa linea, ritorna indietro per iniziarne un'altra e continua questa operazione finché raggiunge il fondo dello schermo. Una volta raggiunta la parte terminale inferiore, ha terminato la sua corsa verso il basso e può ritornare nell'angolo in alto a sinistra con uno spostamento chiamato ritorno verticale. La Figura 2-4 illustra il movimento del fascio di elettroni in una tipica scansione del quadro.

Figura 2-4.

Lo schema del fascio di elettroni utilizzato per la scansione del quadro.



La scansione del quadro deve avvenire molto velocemente, poiché le particelle di fosforo smettono di brillare pochissimo tempo dopo il passaggio del fascio. Per consentire a tutti i fosfori di brillare contemporaneamente sullo schermo, il fascio di elettroni effettua la scansione dell'intero quadro sessanta volte al secondo. La velocità di scansione è conosciuta con il termine di "refresh rate" (velocità di aggiornamento).

I MONITOR A COLORI

Sullo schermo di un monitor a colori esistono tre fosfori di colori diversi: rosso, verde e blu, riuniti in gruppi di tre. Se si

accende un monitor a colori e lo si osserva con una lente di ingrandimento, è possibile vedere brillare le tre diverse particelle colorate (vedi Figura 2-5).

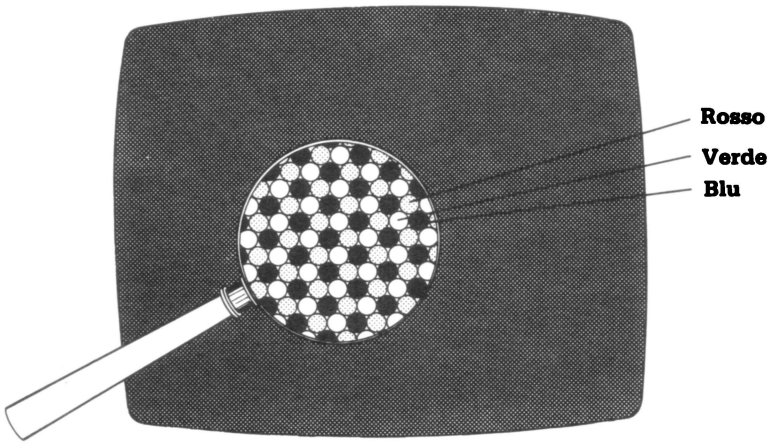


Figura 2-5.

I fosfori sullo schermo di un monitor a colori

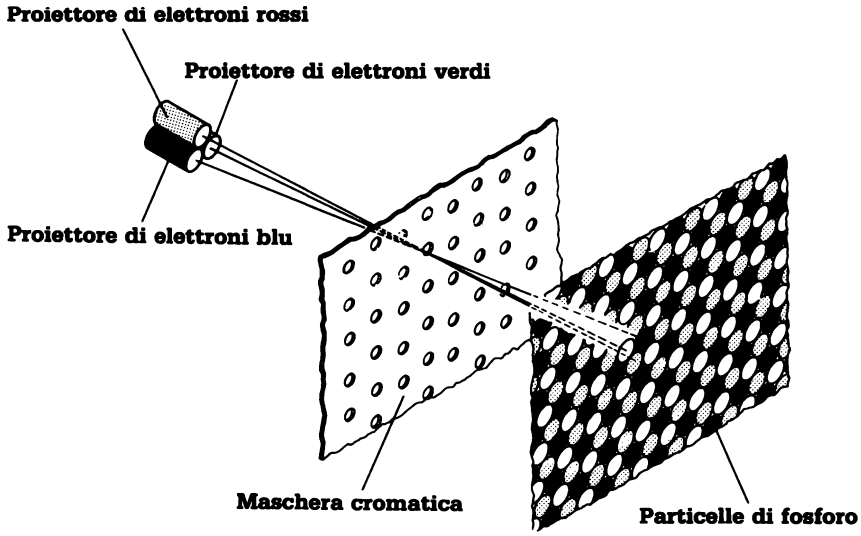
Nella parte posteriore del CRT a colori vi sono tre proiettori di elettroni: uno per i fosfori rossi, uno per i verdi e uno per i blu. Immediatamente dietro lo strato di fosforo del CRT è collocata una maschera cromatica dotata di una griglia con dei minuscoli fori, uno per ciascuna triade di fosfori colorati. Ogni foro concentra i raggi dei tre elettroni mentre questi effettuano la scansione dello schermo, in modo da consentire ad ogni raggio di colpire solo la giusta particella di fosforo colorato in ogni gruppo (vedi Figura 2-6). Nella figura si può osservare come il proiettore di elettroni blu possa colpire soltanto il fosforo blu all'interno della triade; il foro è troppo piccolo perché il raggio colpisca anche i fosfori rossi e verdi. Similmente, il proiettore rosso può colpire solo il fosforo rosso e il proiettore verde solo il fosforo verde.

Il monitor a colori utilizza la stessa scansione del quadro del monitor monocromatico. L'immagine a colori è suddivisa in linee, e i tre fasci di elettroni esplorano la schermo contemporaneamente, muovendosi in direzione orizzontale e verticale sessanta volte al secondo. Invece di produrre sfumature monocromatiche, i tre proiettori di elettroni modificano la loro intensità per ottenere quantità differenti di rosso, di verde e di blu, con proporzioni diverse all'interno dei gruppi di fosfori, allo scopo di visualizzare una varietà di colori sullo schermo.

Per controllare un CRT a colori, il sintonizzatore di un computer o di un televisore deve trasmettere segnali di intensità distinti per ogni proiettore di elettroni, oltre a un segnale che controlli il movimento dei fasci di elettroni. In un monitor RGB questi segnali vengono trasmessi, attraverso cavi separati, direttamente ai proiettori e alle griglie di deflezione.

Figura 2-6.

La maschera cromatica di un CRT a colori concentra ciascuno dei tre fasci di elettroni solo sul fosforo dello stesso colore all'interno di una triade.



In un monitor composito o in un televisore a colori, i segnali vengono uniti ed inviati al monitor attraverso un unico cavo, dopodiché vengono nuovamente separati per essere trasmessi ai proiettori di elettroni e alle aree di deflezione. L'unione e la scissione del segnale deteriorano la qualità dell'immagine. È per questo motivo che i monitor RGB offrono immagini più nitide di quelle dei monitor compositi o dei televisori.

TRASFERIMENTO DI UN'IMMAGINE VIDEO AD ALTRE PERIFERICHE

Le immagini che appaiono sul monitor possono essere viste solo dall'utente o da chiunque gli sieda accanto. Può certamente succedere che l'utente desideri talvolta un più vasto pubblico di spettatori per le sue creazioni grafiche, e abbia quindi la necessità di riprodurle su altri mezzi di comunicazione visiva portatili e accessibili a tutti, come ad esempio videonastri, fotografie o riproduzioni su carta.

Trasmettere immagini video ad altri strumenti non è sempre semplice. In ognuno di essi vi è un diverso tipo di risoluzione, i colori non sempre si combinano come dovrebbero e il senso della proporzione non è sempre lo stesso. Fondamentalmente, la qualità dell'immagine trasmessa a un altro mezzo di comunicazione visiva dipende dalla qualità del software utilizzato per trasferirla, dalla qualità delle apparecchiature adottate per

riprodurla e, infine, dall'abilità dell'operatore e dalla sua conoscenza dello strumento in questione.

VIDEOCASSETTE

La registrazione delle immagini dell'Amiga su una videocassetta è una delle tecniche più semplici di riproduzione della grafica Amiga. L'Amiga può infatti inviare un segnale direttamente dalla porta video (non presente nell'A2000) a un videoregistratore, invece che a un monitor composito, ed è inoltre in grado di gestire più di un monitor alla volta, in modo da consentire all'operatore di visualizzare la grafica anche su un monitor RGB collegato all'apposita porta, mentre registra la stessa immagine su una videocassetta sfruttando la porta video. Il videoregistratore immagazzina le immagini su nastro, cosicché è possibile dare il nastro ad altre persone in possesso dello stesso tipo di videoregistratore, o persino trasmettere le immagini attraverso un ripetitore televisivo, se si è tanto fortunati da potervi accedere.

La videoregistrazione di lavori grafici creata con l'Amiga dà dei risultati veramente eccellenti. È possibile miscelare alle immagini un commento sonoro e trasmettere il tutto con un output via monitor. Gli aspetti negativi sono trascurabili e riguardano soprattutto il fatto che le immagini registrate non risultano di solito all'altezza dell'originale. Inoltre, se si tratta di un'immagine ferma, il videoregistratore può visualizzarla solo finché il nastro non finisce; non è ovviamente possibile mantenere l'immagine sul monitor per un tempo indefinito.

RIPRODUZIONI FOTOGRAFICHE

Un'altra tecnica relativamente semplice per riprodurre le immagini video dell'Amiga consiste nel fotografare lo schermo del monitor. Tuttavia non è possibile puntare la macchina sul quadro, scattare la foto e aspettarsi risultati soddisfacenti. È necessario adottare alcune misure precauzionali per ottenere fotografie di buona qualità.

Lo schermo deve essere fotografato in un ambiente completamente oscurato, in modo da impedire che la luce esterna si rifletta sullo schermo sbiadendo l'immagine. Poiché la scansione standard del quadro viene effettuata ogni 1/60 di secondo, è necessario assicurarsi che la velocità dell'otturatore della macchina fotografica non sia superiore a 1/30 di secondo. Si rivela inoltre necessario un cavalletto per tenere ferma la macchina mentre si scatta la foto. Con una velocità più elevata la scansione del quadro viene fotografata a processo solo parzialmente completo. Ad esempio, con una velocità di 1/120 di secondo si fotografa solo metà dell'immagine sullo schermo.

Le stampe fotografiche del video hanno l'ovvio vantaggio di essere facilmente trasportabili. Possono essere spedite per posta o appese alla parete, e possono essere guardate facilmente. Non è infatti necessario perdere tempo per collegarle a una presa di corrente, accenderle e caricarle. Come ogni altro tipo di riproduzione fotografica, possono essere utilizzate in mille modi diversi.

Le fotografie a colori presentano però alcuni svantaggi. Non sempre riproducono i colori del monitor con assoluta fedeltà e, inoltre, non sono luminose quanto il monitor. Un sistema per migliorare la riproduzione dei colori consiste nel fotografare con pellicole per diapositive. Queste offrono infatti il vantaggio di risultare luminose come lo schermo del monitor quando vengono proiettate. Fotografie e diapositive hanno però in comune il problema della distorsione, causato dal fatto che lo schermo del monitor presenta una lieve curvatura. Quando lo si osserva direttamente, le immagini risultano piatte, perché il nostro cervello provvede automaticamente a regolare le distorsioni. L'immagine fotografata è invece trasferita su un dispositivo bidimensionale che evidenzia la curvatura. L'immagine risulta perciò leggermente curvata al centro.

RIPRODUZIONI SU CARTA

Le immagini video sono comunemente riprodotte con una stampante in grado di riportare la grafica su carta. È un sistema semplice, rapido ed economico (una volta acquistata la stampante). Non è necessario attendere giorni perché il fotografo restituisca le fotografie sviluppate. Conoscere i risultati è questione di pochi minuti. Le immagini stampate, inoltre, eliminano il problema della curvatura riscontrato per le fotografie dello schermo.

La qualità delle immagini stampate varia considerevolmente da una stampante all'altra. Nel migliore dei casi, un'immagine a colori riprodotta su carta è molto simile all'originale del video, mentre nel peggiore la riproduzione fornisce ottime motivazioni per rivendere la stampante.

Nel processo di trasferimento dell'immagine su carta, la componente più critica che va a intaccare la fedeltà della riproduzione è probabilmente la qualità della stampante stessa. La conversione dell'immagine video in dati che la stampante possa comprendere e successivamente riprodurre costituisce un secondo aspetto del problema. Per questo motivo la qualità del software che comanda la stampante è di importanza fondamentale: gli ostacoli da superare sono infatti parecchi.

La risoluzione

Il fatto che la maggior parte delle stampanti sia in grado di riprodurre un'immagine con un maggior numero di dettagli

(una risoluzione più elevata) rispetto all'immagine visualizzata sul monitor, costituisce uno dei primi problemi della trasformazione delle immagini video in stampe su carta. Una stampante a impatto di media qualità può stampare 900 punti su ogni linea e 1200 su ogni colonna di un foglio di formato 8,5 per 11 pollici. L'immagine più dettagliata che l'Amiga è in grado di visualizzare è costituita da 640 pixel orizzontali e 400 (512 nel nostro sistema televisivo) verticali. Si tratta di un'immagine di risoluzione inferiore rispetto all'immagine stampata di 900 per 1200 punti. Per riprodurre un'immagine su carta, l'Amiga deve quindi convertirla.

Il modo più semplice per convertire la risoluzione del video si basa sull'utilizzo di blocchi di punti d'inchiostro per rappresentare ogni elemento dell'immagine su video (detto pixel). In un'immagine video di 320 per 200, ogni pixel può essere riprodotto su carta con un blocco corrispondente di 3 per 3 punti, in modo da dilatare l'immagine di 320 per 200 a una riproduzione di 960 per 600 punti di inchiostro. Il problema della conversione di pixel in blocchi sta nel fatto che i contorni dell'immagine su carta risultano seghettati. Nella Figura 2-7 è riprodotta una linea diagonale così come appare sul monitor e, a fianco, il risultato ottenuto con la stampante. Quest'ultima presenta dei contorni definiti in maniera poco precisa.

Figura 2-7.

"Seghettature" ottenute riproducendo su carta un'immagine video

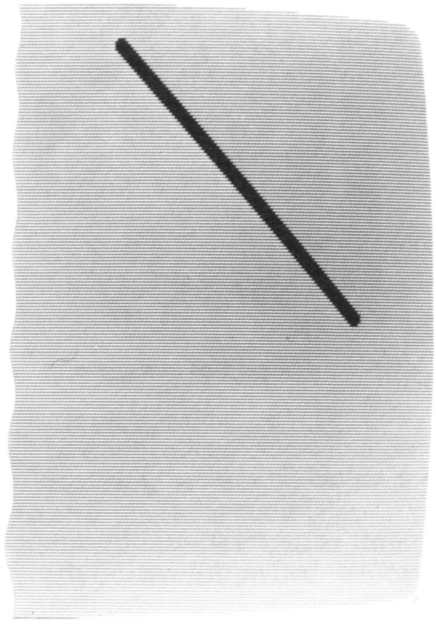


Immagine dello schermo

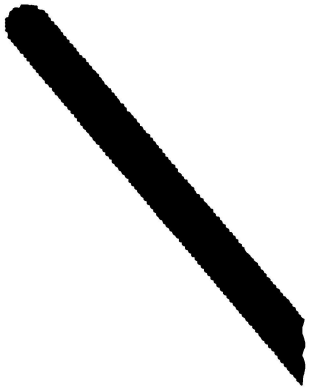


Immagine stampata

Il software delle stampanti di alta qualità, invece, pur utilizzando sempre più di un punto di stampa per rappresentare un pixel, non utilizza il sistema dei blocchi, bensì uno schema più razionale che consente di appiattire le seghettature (termine usato per intendere l'effetto "a scalino" nelle linee curve e diagonali). Questo software sfrutta al meglio la più alta risoluzione della stampante, come si può vedere nella Figura 2-8. L'eliminazione delle seghettature attuata con questo sistema è definita "anti-aliasing". L'immagine ha così un aspetto migliore, ma il software deve intuire le intenzioni dell'operatore. Se l'utente intende tracciare di proposito immagini seghettate, potrebbe succedere, in una stampa che utilizzi il software anti-aliasing, che le frastagliature vengano smussate contro la volontà dell'operatore.

Figura 2-8.

L'anti-aliasing liscia le seghettature di una linea diagonale stampata.

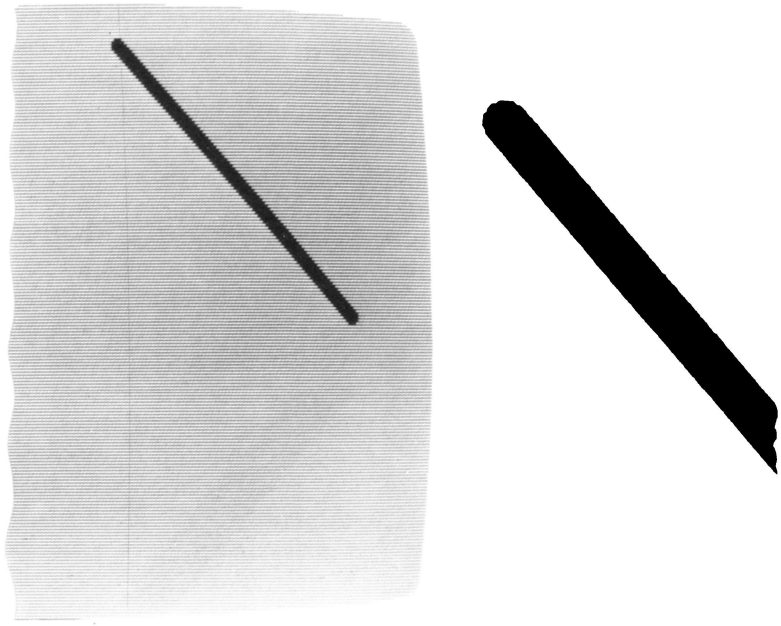


Immagine dello schermo **Immagine stampata**

La conversione di immagini a colori in stampe in bianco e nero

La maggior parte delle stampanti per computer non stampa a colori, ma solo in bianco e nero. Per convertire i colori di un'immagine in tratti in bianco e nero il software offre due possibilità: riprodurre l'immagine unicamente in bianco e nero, oppure utilizzare sfumature di grigio contenute nella scala dei grigi compresa fra il nero e il bianco.

La riproduzione dell'immagine in bianco e nero è la modalità

più semplice per convertire i colori del video. Il software si limita a stabilire un punto di riferimento per la luminosità. Qualsiasi pixel dell'immagine video più luminoso del punto di riferimento è stampato completamente in bianco, qualsiasi pixel più scuro viene invece riprodotto in nero. L'evidente svantaggio della stampa strettamente in bianco e nero è la perdita delle finezze delle sfumature dell'immagine originale. Uno dei vantaggi, invece, risiede nella possibilità di migliorare l'aspetto di un'immagine dai colori poco appropriati, riproducendola totalmente in bianco e nero.

Per convertire i colori in sfumature di grigio, il software utilizza matrici di punti neri allo scopo di ottenere diverse gradazioni. Ogni gradazione è rappresentata da una matrice diversa. La Figura 2-9 illustra alcuni ingrandimenti di matrici di sfumature di grigio. Per passare dai colori del video al grigio, il software interpreta i pixel luminosi come sfumature chiare di grigio e i pixel scuri come tonalità più scure, e quindi le stampa usando le matrici corrispondenti.

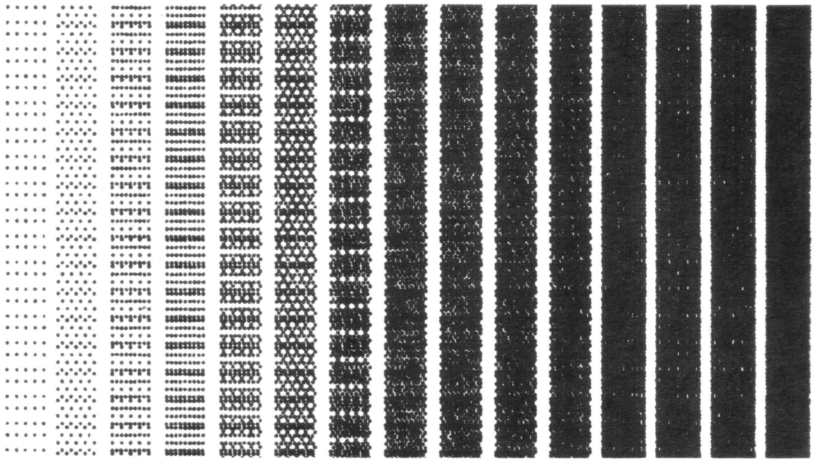


Figura 2-9.

Matrici delle sfumature di grigio (ingrandimento).

Stampa a colori

Le stampanti a colori non devono interpretare i colori del video come sfumature di grigio, ma devono comunque risolvere altri problemi complicati, il più importante dei quali è costituito dal fatto che la stampante si serve di un insieme di colori fondamentali che non corrispondono ai colori base del video. Per creare tinte intermedie, le stampanti a colori usano il cyan, il giallo e il magenta in diverse combinazioni, e spesso aggiungono nero per dare all'immagine maggior contrasto.

La conversione diretta del rosso, del verde e del blu in cyan, giallo e magenta, non presenta alcuna difficoltà se non per un fattore importante: i colori fondamentali sullo schermo del monitor, a differenza di quelli della stampante, possono variare di intensità. I colori primari di una stampante possiedono un solo grado di intensità, perché un punto di inchiostro, diversamente da un fosforo, non può variare di luminosità. Quando una stampante a colori combina due colori fondamentali, ne deriva solo una terza tinta, non una di più. Per questo motivo la maggior parte delle stampanti a colori possiede solo sei differenti tinte uniformi: cyan, giallo, magenta, verde (associazione di cyan e giallo), rosso (giallo e magenta) e viola (cyan e magenta).

Per approssimare le migliaia di colori che il video può riprodurre, le stampanti a colori utilizzano un processo chiamato "dithering" (tremolio). Il termine, che suona alquanto curioso, indica semplicemente che la stampante ricopre con punti di un dato colore un settore uniformemente dipinto con un'altra tinta. Se osservato da una certa distanza, il dithering produce un nuovo colore. Ad esempio, un settore completamente rosso cosparso di punti neri risulta rosso scuro.

Questo procedimento consente alla stampante di disporre di un'ampia varietà di colori, ma la sua gamma cromatica resta limitata se raffrontata a quella del video. Ne consegue che un'immagine Amiga trasferita a una stampante non mostra le più sottili sfumature cromatiche. Come risultato del dithering, inoltre, i colori riportati su carta presentano una struttura che conferisce loro un aspetto differente da quello delle corrispettive tinte sul video.

LE ECCEZIONALI CARATTERISTICHE GRAFICHE DELL'AMIGA

Dopo aver visto come le immagini vengono create sul monitor dell'Amiga, recepite dall'occhio umano e trasferite ad altre periferiche, si possono comprendere meglio i risultati delle particolari caratteristiche grafiche dell'Amiga. L'Amiga è stato progettato per essere un eccezionale computer grafico, e la potenza del suo hardware, unita alle routine grafiche di cui dispone il suo software sistema, gli consentono di creare immagini eccezionali e raffinate.

Le pagine seguenti descrivono le caratteristiche grafiche di qualsiasi computer Amiga che funzioni con il proprio software sistema (quello contenuto su Kickstart e Workbench). Anche se non sarete in grado di utilizzare immediatamente tutte le caratteristiche della macchina (a meno che non siate un

esperto programmatore in grado di usare le Exec libraries e devices), la maggior parte del potenziale Amiga può essere sfruttato attraverso programmi applicativi come Deluxe Paint o linguaggi di programmazione come l'Amiga BASIC, come verrà descritto nei capitoli successivi. Col passare del tempo, più aumentano le pubblicazioni di software per l'Amiga, maggiori sono le possibilità che nuovi programmi applicativi mettano a disposizione dell'utente un numero sempre più grande di caratteristiche peculiari dell'Amiga.

I DETTAGLI

Come la maggior parte degli altri microcomputer, l'Amiga crea immagini sullo schermo utilizzando i pixel, minuscoli punti quadrati che non consentono una perfetta definizione dei contorni delle immagini. Creare una figura con i pixel è come costruire una casa con dei mattoni: sono disponibili solo angoli squadrati. Il segreto per eliminare le seghettature sui contorni consiste nell'usare pixel più piccoli: da una certa distanza, le scalinature sono indistinguibili. L'Amiga offre una scelta fra quattro diverse dimensioni di pixel, che permettono di ottenere figure più o meno dettagliate ed uniformi. I diversi pixel sono rappresentati nella Figura 2-10.

La dimensione del pixel utilizzato determina la risoluzione dello schermo. Più piccoli sono i pixel, più alta è la risoluzione e maggiori sono i dettagli della figura riprodotta sullo schermo. Più grandi sono i pixel, più bassa è la risoluzione e meno precisa è la figura visualizzata.

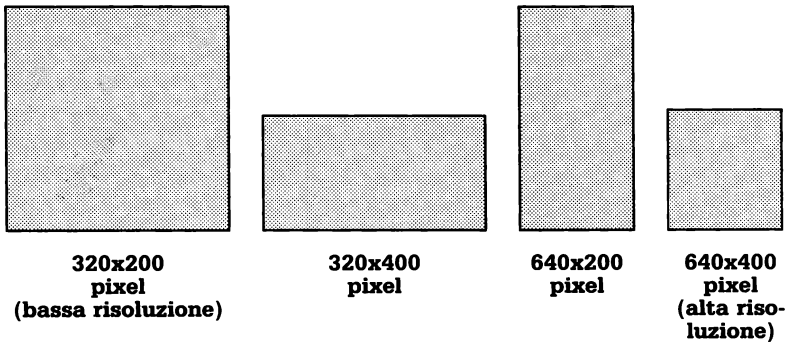


Figura 2-10.

Le quattro diverse dimensioni dei pixel della grafica Amiga, notevolmente ingranditi.

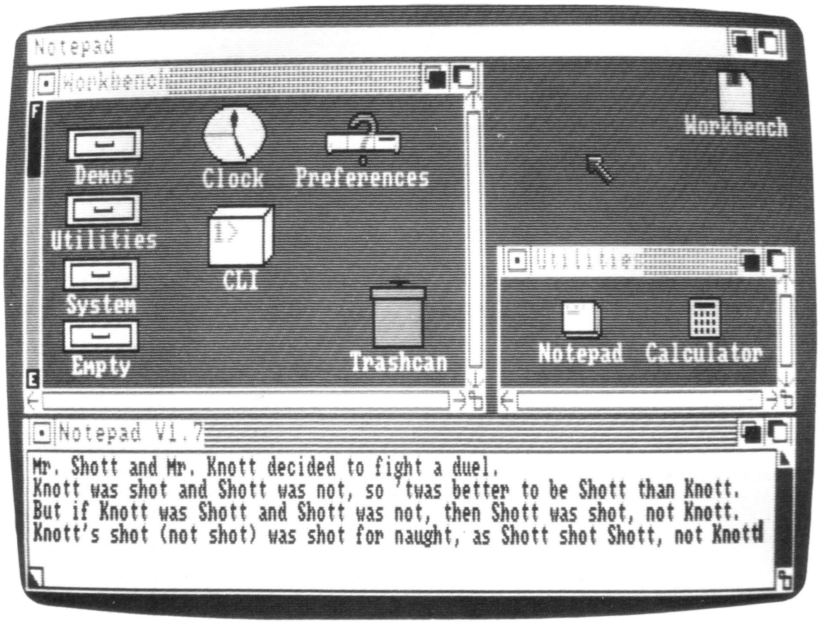
La risoluzione più bassa dello schermo dell'Amiga è 320 per 200 (256 nelle versioni europee della serie Amiga), vale a dire 320 pixel in orizzontale e 200 in verticale. L'immagine della Figura 2-11 (riportata nella pagina successiva) è realizzata utilizzando lo schermo a bassa risoluzione. Guardando da vicino si vede chiaramente che la forma dei pixel è piuttosto squadrata, mentre da lontano i particolari appaiono molto più definiti e i contorni non risultano frastagliati.

Figura 2-11.

Un'immagine realizzata con lo schermo Amiga a bassa risoluzione 320 per 200 pixel *("Venere" per gentile concessione di Avril Harrison).*

Figura 2-12.

Una schermata Amiga a media risoluzione 640 per 200 pixel: lo schermo del Workbench.



Nelle due risoluzioni intermedie dello schermo dell'Amiga, il pixel a bassa risoluzione è tagliato a metà per creare un pixel rettangolare. Nella risoluzione 320 per 400 (512 nelle versioni europee), il pixel a bassa risoluzione è tagliato a metà orizzontalmente, per dare l'idea di un mattone appoggiato sulla faccia più lunga. Nella risoluzione 640 per 200 (256 nelle versioni europee) il pixel è tagliato a metà verticalmente e appare come un mattone in piedi. Questo secondo tipo di pixel è molto utile per stampare testi, in quanto permette l'inserimento di un numero maggiore di caratteri sullo schermo. Nella Figura 2-12 (della pagina precedente) sono rappresentati sia testi che disegni realizzati su uno schermo Amiga a risoluzione 640 per 200.

Nello schermo ad alta risoluzione il pixel ha forma quadrata ed è grande esattamente un quarto del pixel a bassa risoluzione. Lo schermo ad alta risoluzione misura 640 pixel in orizzontale e 400 (512 nelle versioni europee) pixel in verticale. Con unità così piccole i particolari sono molto raffinati e non si notano le seghettature dei contorni. Nella Figura 2-13 è riprodotta un'immagine Amiga ad alta risoluzione. I particolari sono così dettagliati che si possono notare persino i peli sul muso del mandrillo.

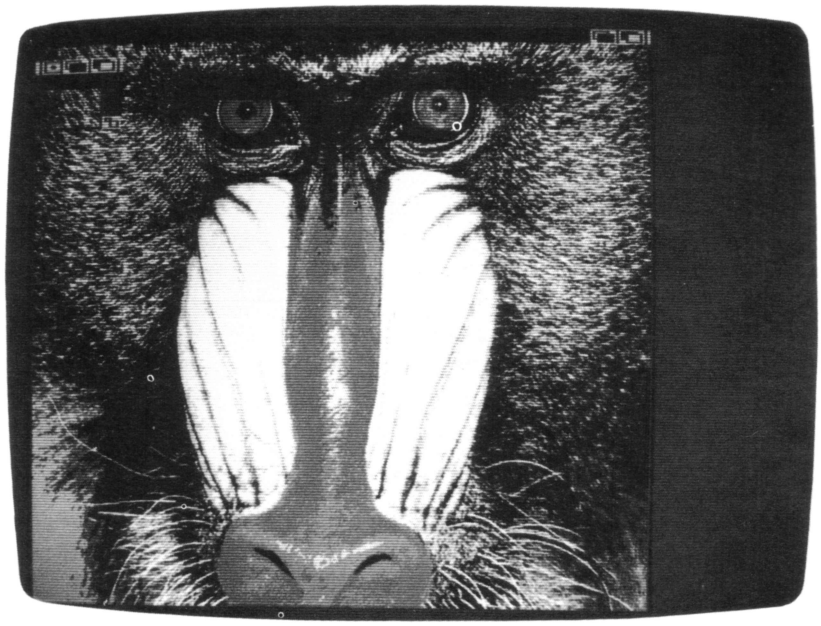


Figura 2-13.

Un'immagine creata utilizzando la risoluzione 640 per 400 pixel

Incidenza della risoluzione sulla memoria

Perché ci sono diverse risoluzioni a disposizione dell'Amiga? Non sarebbe meglio usare sempre uno schermo ad alta risoluzione? La risposta a questi quesiti è data dalla quantità di memoria richiesta da ogni risoluzione. L'Amiga memorizza i dati di un'immagine, e più alta è la risoluzione maggiore è lo spazio di RAM necessario per gestirla. Ad esempio, con una figura a bassa risoluzione i pixel da memorizzare sono solo 64000 (320x200), mentre in una figura ad alta risoluzione il numero di pixel da memorizzare è di 256000, quattro volte il numero di pixel di un'immagine a bassa risoluzione.

Con quattro diverse risoluzioni a disposizione, è possibile scegliere quella più adatta alle esigenze del caso specifico senza occupare troppa RAM. È anche possibile usare i pixel di diverse dimensioni come veri e propri strumenti grafici, al fine di ottenere risultati raffinati o volutamente scadenti, nello stesso modo in cui un pittore sceglie diversi tipi di carta per un acquerello. Nei prossimi capitoli verrà spiegato come selezionare la risoluzione desiderata.

Miscelando le risoluzioni

Se si desidera utilizzare contemporaneamente diverse risoluzioni sul monitor, l'Amiga offre la possibilità di dividere il quadro in fasce orizzontali chiamate schermi (da non confondere con lo schermo del monitor), ognuno con la propria risoluzione. Intuition, l'interfaccia utente, permette di realizzare diverse risoluzioni sullo stesso quadro dividendo lo schermo in più sezioni. Muovendo il puntatore del mouse verso il basso, è possibile visualizzare gli schermi che si trovano più sotto; muovendo l'indicatore verso l'alto, invece, gli schermi inferiori spariscono. La Figura 2-14 (riportata nella pagina sucessiva) mostra una serie di schermi a diversa risoluzione riuniti in un unico quadro.

La maggior parte dei programmi per Amiga, come ad esempio il Workbench, utilizza un solo schermo con una risoluzione predeterminata. Alcuni programmi, come Deluxe Paint, offrono la possibilità di scegliere la risoluzione, ma poi la utilizzano per l'intero schermo. Altri programmi, fra cui Deluxe Video, utilizzano due o più schermi a diverse risoluzioni che si possono anche spostare sulla superficie del monitor. Per creare schermi propri è necessario utilizzare linguaggi di programmazione come l'Amiga BASIC o il C (nel quarto capitolo si parlerà più ampiamente degli schermi e di come crearli).

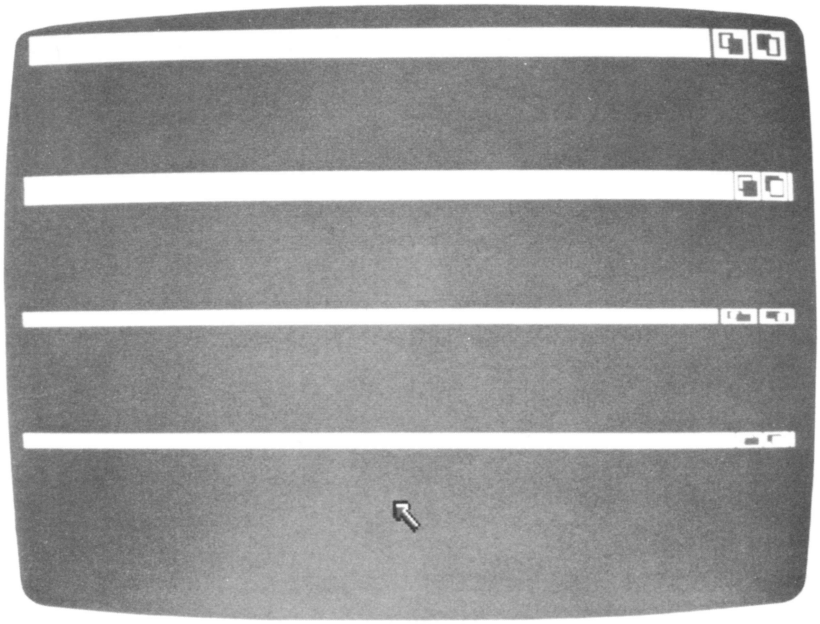


Figura 2-14.

Uno schermo Amiga con le quattro diverse risoluzioni visualizzate contemporaneamente.

LA PALETTE DEI COLORI DELL'AMIGA

La risoluzione è solo uno degli aspetti che caratterizzano le immagini dell'Amiga. L'altro aspetto fondamentale è il colore. La risoluzione conferisce all'immagine altezza e larghezza, il colore può aggiungere l'idea della profondità.

L'Amiga può creare sullo schermo un massimo di 4096 colori, usando le varie combinazioni di rosso, verde e blu. Di solito non è possibile visualizzare tutti questi colori contemporaneamente. Come per le risoluzioni, un numero più elevato di colori richiede un maggiore spazio in memoria per l'immagine. Per mantenere la richiesta di memoria entro limiti ragionevoli, l'Amiga pone un limite massimo di 32 colori per schermo (tranne che per gli schermi a risoluzione 320 per 400 e 640 per 400, dove il massimo consentito è 16). Questo sistema permette di avere a disposizione una vasta gamma di colori senza utilizzare uno spazio di memoria tale da non poter più far girare alcun programma.

L'Amiga utilizza un sistema ingegnoso affinché la varietà dei 32 colori risulti piuttosto flessibile. La macchina colora le immagini come nelle figure a colori numerati con cui giocano i bambini; a ogni pixel dell'immagine viene assegnato un

numero compreso fra 0 e 31. Poi l'Amiga colora ogni pixel con il colore corrispondente al numero che contrassegna il pixel stesso.

I colori che corrispondono ai vari numeri sono memorizzati in 32 registri di colore separati, anch'essi numerati da 0 a 31. Questi registri sono piccole sezioni individuali di memoria che memorizzano il colore come combinazione di diverse quantità di rosso, verde e blu. L'Amiga invia queste combinazioni al monitor, che le utilizza per visualizzare le diverse tonalità, colorando ogni pixel col colore del corrispondente registro. Ognuno dei registri di colore può memorizzare una delle 4096 combinazioni possibili di rosso, verde e blu che l'Amiga è in grado di produrre, in modo da offrire una gamma cromatica molto varia per creare qualsiasi combinazione di colori serva per le immagini riprodotte sullo schermo.

Il vantaggio principale dei registri di colore è senza dubbio il fatto che essi offrono una grande varietà di tonalità senza occupare troppa RAM; ma c'è anche un altro vantaggio. Cambiando il colore in uno solo dei registri, tutti i pixel contrassegnati dal numero del registro modificato cambiano automaticamente colore. Tale uso dei registri di colore permette di verificare senza difficoltà le diverse combinazioni di tonalità.

Ad esempio, utilizzando Deluxe Paint per disegnare sullo schermo un tessuto stampato con figure verdi su sfondo rosso, si può provare a vedere l'effetto dei disegni verdi su uno sfondo arancione. Non è necessario ritoccare il disegno colorando lo sfondo di arancione; è sufficiente, per cambiare il colore sullo schermo, selezionare il colore dello sfondo e cambiarlo da rosso ad arancione, utilizzando i cursori per la definizione dei colori situati a lato della palette (le modalità verranno descritte nel prossimo capitolo). Ogni pixel rosso sullo schermo viene automaticamente mutato in arancione e i risultati sono immediatamente visibili.

L'Amiga può visualizzare tutti i 4096 colori sullo schermo contemporaneamente, utilizzando un procedimento chiamato "Hold and Modify" (noto con la sigla HAM), che stende i colori orizzontalmente creando sottilissime sfumature. Si può ad esempio utilizzare il sistema HAM per realizzare giochi di luce e ombra su un vaso rosso tondeggiante, in modo da farlo apparire tridimensionale. I punti in cui il vaso è colpito dalla luce vengono colorati di rosso brillante, che viene gradatamente sfumato in toni più scuri nei punti in ombra. Alla fine, il vaso avrà un aspetto liscio e lucente. La Figura 1-15 (riportata nella pagina successiva) mostra un esempio di immagine creata col procedimento HAM.

Figura 2-15.

Un'immagine Amiga che utilizza il metodo "Hold and Modify" (HAM) per riprodurre i giochi di ombre *(per gentile concessione della NewTek e di Mitchell Lopes).*

Al momento della stesura di questo libro, ad eccezione di Digi-Paint non esistono programmi grafici che utilizzano l'HAM, anche se è probabile che ne verranno creati alcuni molto presto. Per ora esistono alcuni digitalizzatori video per l'Amiga che convertono le immagini di una telecamera o di altre fonti video in una figura HAM, riprodotta tramite l'utilizzazione di raffinatissime sfumature.

DISEGNO DI FIGURE

L'Amiga è dotato di particolari routine, contenute nella sua libreria grafica, che facilitano la creazione di immagini sullo schermo. Queste routine grafiche creano tutte le componenti di una figura: possono infatti tracciare linee, riempire alcune sezioni con colori o matrici grafiche particolari, copiare un particolare dell'immagine in un'altra parte del monitor, modificare la colorazione e svolgere altre importanti funzioni.

Le routine grafiche si servono di una sezione particolare del chip Agnus (uno dei tre coprocessori) chiamata blitter. Blitter è l'abbreviazione di "bit-mapped block transferrer" e indica la capacità di gestire rapidamente grandi quantità di dati presenti in memoria. Quando questi dati si trovano nella sezione di visualizzazione grafica dell'Amiga, il blitter può disegnare le figure molto velocemente; in questo modo i risultati non si fanno attendere a lungo.

Le routine grafiche sono usate frequentemente nel software per l'Amiga. Tutti i comandi grafici dell'Amiga BASIC e le

funzioni di programmi come Deluxe Paint utilizzano routine grafiche per svolgere quanto loro richiesto. Il Workbench utilizza invece tali routine per disegnare icone e finestre sullo schermo.

CREAZIONE DI TESTI

Nel software sistema dell'Amiga, le librerie creano anche i caratteri, in modo da consentire l'inserimento di testi. Per farlo utilizzano un sistema molto versatile, simile ai registri di colore, ma invece di memorizzare in ogni registro una diversa combinazione di rosso, verde e blu, vi immagazzinano il disegno di un carattere. L'intero set di caratteri contenuto in memoria (lettere dell'alfabeto, numeri, punteggiatura, simboli matematici e altri segni speciali) viene chiamato fonte.

Quando l'utente scrive con la tastiera, l'Amiga utilizza i disegni della fonte che ha in memoria per riprodurre i caratteri digitati sullo schermo. Quando si cambia la fonte, l'Amiga modifica i disegni di tutti i registri, cosicché tutti i caratteri che appariranno sullo schermo utilizzeranno la grafica della nuova fonte. Una fonte può essere lineare ed essenziale, un'altra ricercata ed elegante. Ad esempio, in un programma di elaborazione testi l'Amiga può utilizzare una fonte per il titolo e un'altra per il testo. La Figura 2-16 illustra alcune delle fonti utilizzate dall'Amiga.

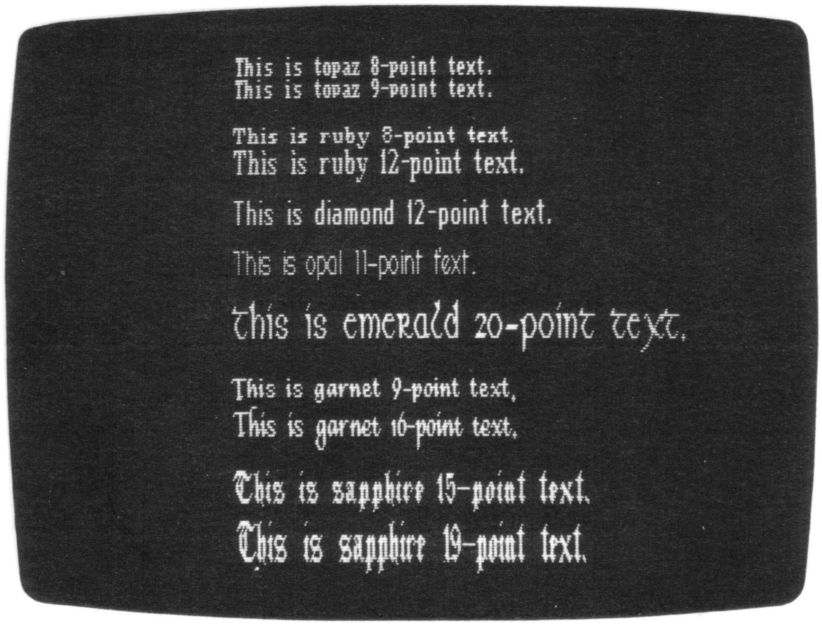


Figura 2-16.

Alcune delle fonti di caratteri dell'Amiga.

Le diverse fonti sono conservate su disco; quando l'Amiga ne ha necessità, vengono trasferite nella RAM e utilizzate per creare il testo sul monitor. Sono già disponibili dischetti-fonti che consentono di caricare nuovi set di caratteri da floppy disk e richiamarli nel programma che si sta utilizzando.

I programmatori in C o in linguaggio Assembly possono creare una fonte propria, disegnando il modello di ogni carattere e memorizzando i dati su disco. Questo sistema è molto utile a chi ha bisogno dei caratteri dell'alfabeto russo, greco, ebraico o di altre lingue che utilizzano simboli alfabetici particolari. Lo stesso discorso vale per la creazione di particolari simboli logici o matematici. I non programmatori devono invece rivolgersi all'acquisto di software che consenta loro di creare fonti personali.

Quando la fonte è caricata in memoria, l'Amiga può utilizzare il suo software sistema per alterare i caratteri. Ad esempio, può raddoppiarne la larghezza, sottolinearli, scriverli in corsivo o in grassetto, invertirne i colori. Può anche colorarli con uno dei 4096 colori disponibili. Alcuni di questi effetti sono reperibili in programmi di word processing come Textcraft ed in programmi grafici con la possibilità di inserimento di testi come Deluxe Paint.

L'Amiga controlla anche lo sfondo dei caratteri. Come sfondo può usare le immagini già esistenti sullo schermo oppure creare un background contrastante. I caratteri su sfondo contrastante assomigliano a strisce di lettere incollate su un telegramma. I caratteri che utilizzano uno sfondo già esistente si amalgamano con il resto della figura. Nella Figura 2-17 sono riportati entrambi i casi.

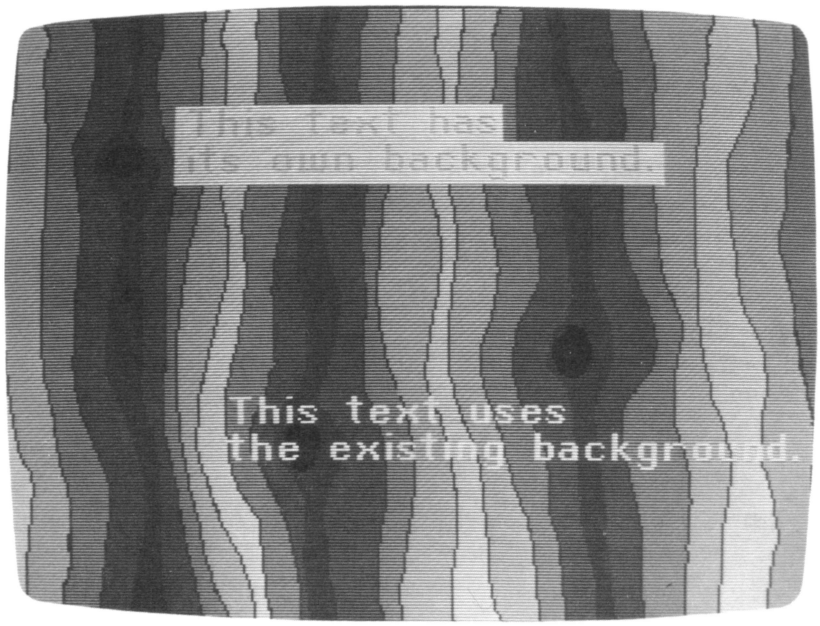


Figura 2-17

I caratteri in alto sono stati disposti sul disegno sottostante con uno sfondo contrastante. I caratteri in basso utilizzano come sfondo il disegno già presente sullo schermo.

SCORRIMENTO DELLE IMMAGINI

L'Amiga dispone di molte caratteristiche attualmente accessibili solo ai programmatori esperti che utilizzano il linguaggio Assembly o il C. Fra queste caratteristiche vi è quella che permette all'elaboratore di memorizzare una figura che risulti troppo alta o troppo larga per essere visualizzata interamente sullo schermo. In questo caso, l'Amiga mostra solo una parte della figura, utilizzando lo schermo come una sorta di finestra. La Figura 2-18 illustra il funzionamento di questo sistema.

Anche se sullo schermo appare solo una sezione dell'immagine, l'Amiga è in grado di riprodurre ogni singola parte della figura e quindi di mostrarla per intero, anche se in fasi successive. Si può passare velocemente da una sezione all'altra oppure far scorrere lentamente la figura sullo schermo. L'impressione è che la figura scivoli dietro il quadro. L'Amiga può far scorrere l'immagine a qualsiasi velocità, non solo verso l'alto o verso il basso, a destra o a sinistra, ma anche in direzione diagonale.

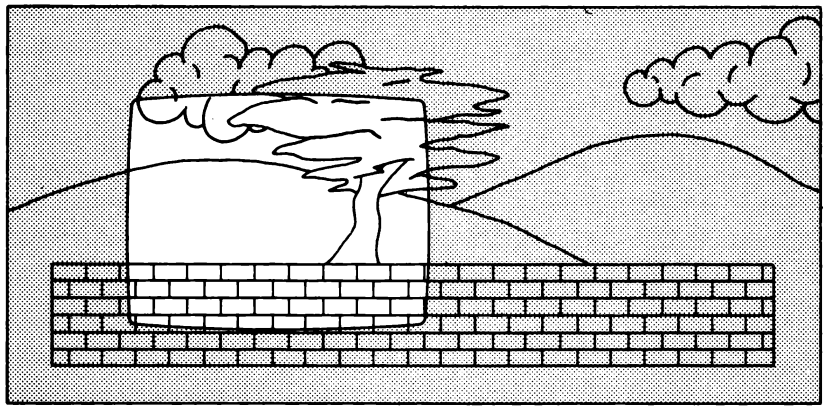


Figura 2-18.

Lo schermo dell'Amiga funziona come una finestra che mostra una parte di un'immagine molto grande contenuta in memoria.

Un buon esempio dell'utilità dello scroll (scorrimento) può essere realizzato scrivendo un programma in C che disegni una mappa di notevoli dimensioni e la inserisca nella memoria dell'Amiga. Lo schermo mostra solo una parte della mappa, ma può essere spostato a nord, a sud, a est o a ovest per vedere sezioni che altrimenti resterebbero fuori quadro.

SOVRAPPOSIZIONE DI DUE IMMAGINI

Nella grafica per computer, l'Amiga è all'avanguardia per un'altra caratteristica: può memorizzare contemporaneamente due figure e poi sovrapporle. Quando l'Amiga sovrappone due immagini, fa di una lo sfondo e dell'altra il primo piano. Solitamente, la figura in primo piano copre parte dello sfondo, ma per alcuni pixel è possibile utilizzare un colore "trasparente". Questi pixel particolari lasciano intravedere l'immagine sullo sfondo. La Figura 2-19 illustra i risultati del procedimento descritto.

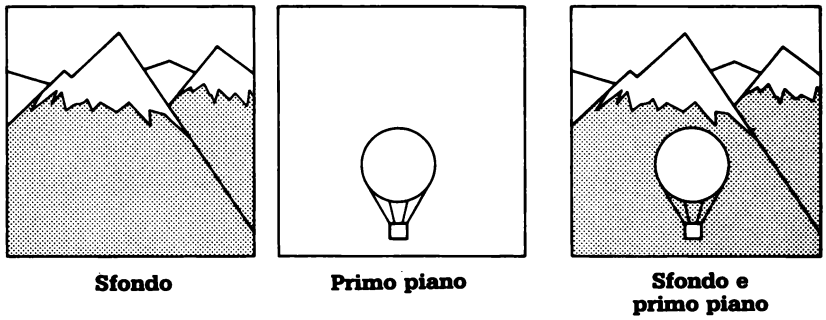


Figura 2-19.

Sulla sinistra, le due figure memorizzate dall'Amiga sono rappresentate singolarmente. Sulla destra, la prima è sovrapposta alla seconda, come apparirebbe sul monitor.

Sovrapponendo una figura all'altra, risulta più semplice cambiare le immagini che appaiono sullo schermo. L'immagine sullo sfondo può essere utilizzata come lo scenario di un teatro: viene disegnata una sola volta e non subisce più modifiche. Qualsiasi elemento che dovrà poi cambiare o muoversi deve essere disegnato in primo piano. Dal momento che l'Amiga non deve continuamente cambiare l'immagine retrostante come avviene per il primo piano, è molto più semplice aggiornare la raffigurazione, dovendo la macchina ridisegnare solo il primo piano.

Anche se quella di sovrapporre immagini è una tecnica generalmente accessibile solo ai programmatori più esperti che scrivono in C o in Assembly, anche l'utente inesperto può far uso di immagini sovrapposte ricorrendo al programma di animazione Deluxe Video (trattato nell'undicesimo capitolo).

STAMPA DI UN'IMMAGINE AMIGA

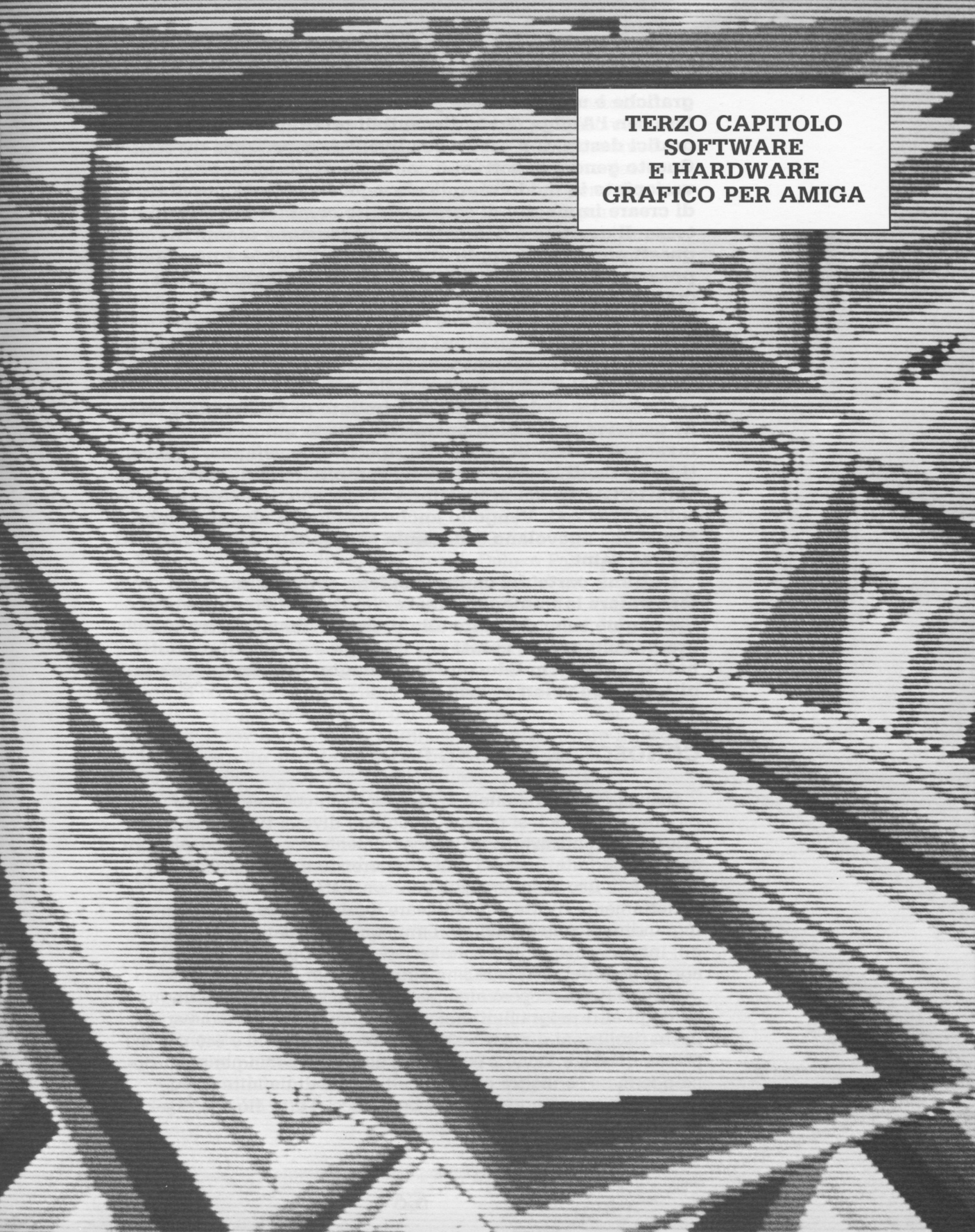
Come spiegato in precedenza, non è facile riprodurre su carta un'immagine video. Tale processo richiede una conversione di risoluzione e di colori, conversione che varia a seconda della stampante. Il software che si occupa della conversione viene chiamato "printer driver".

La sezione Preferences del Workbench offre una scelta tra numerosi printer driver che il sistema può utilizzare per adattarsi a tutte le diverse stampanti che l'utente può collega-

re all'Amiga attraverso la porta seriale o la porta parallela. L'utente può quindi scegliere il driver adatto alla stampante da lui collegata all'Amiga. Quando vengono realizzati nuovi modelli di stampanti, i produttori e la Commodore realizzano e pubblicano nuovi printer driver. Questi nuovi driver possono essere caricati attraverso lo schermo Preferences per permettere all'Amiga di usufruire delle nuove stampanti.

È molto più vantaggioso disporre di un printer driver attivo nel software sistema, piuttosto che doverne creare uno per ogni programma applicativo. Alcuni sistemi di computer costringono il programmatore a crearsi un proprio driver nel caso desideri usare una stampante. Questo fatto costringe gli utenti a essere molto cauti quando devono acquistare un programma che prevede l'uso della stampante: chi dice che quel programma funziona con la loro stampante? Avendo già settato nella sezione Preferences un printer driver adatto alla vostra stampante, sarà il software dell'Amiga stesso a far funzionare la stampante, in modo che anche il software acquistato si adatti automaticamente alla stampante posseduta.

Abbiamo visto le tecniche fondamentali utilizzate dall'Amiga per creare e visualizzare composizioni grafiche. I prossimi quattro capitoli sono dedicati alla creazione di immagini col programma applicativo Deluxe Paint e all'inserimento di elementi di grafica nei programmi in Amiga BASIC di vostra ideazione.



TERZO CAPITOLO SOFTWARE E HARDWARE GRAFICO PER AMIGA

Creare immagini con un programma di applicazioni grafiche è una delle cose più divertenti che si possano fare con l'Amiga. Sono disponibili programmi applicativi grafici destinati a utenti con livelli di esperienza diversi. Questo genere di software consente di disegnare figure variopinte in modo semplice e veloce e offre la possibilità di creare immagini di sorprendente complessità e raffinatezza. Il capitolo è dedicato alla descrizione di uno di questi programmi applicativi, Deluxe Paint, e a una breve presentazione di altri due programmi analoghi: Graphicraft e Aegis Images.

Deluxe Paint, prodotto e distribuito dalla Electronic Arts, è uno dei programmi grafici applicativi più versatili che esistano sul mercato. In questo capitolo imparerete a usufruire nel modo migliore delle sue avanzatissime caratteristiche. Deluxe Paint è corredato di un manuale che ne descrive le funzioni operative. Perciò qui di seguito, invece di cadere in un'inutile ripetizione, vengono illustrati i vari modi in cui è possibile combinare le diverse caratteristiche del programma allo scopo di ottenere risultati concreti. Vengono anche presentati alcuni suggerimenti che agevolano il normale lavoro dell'utente con Deluxe Paint.

L'ultima parte del capitolo spiega come stampare o fotografare le immagini create con Deluxe Paint. Le ultime pagine sono dedicate alla presentazione di altri due programmi grafici creati per l'Amiga e di alcuni prodotti hardware che riproducono le immagini conservandone la nitidezza e i colori vivaci.

CONOSCERE DELUXE PAINT

Deluxe Paint ha molte caratteristiche in comune con gli altri programmi grafici: si possono scegliere diversi tipi di pennello selezionabili attraverso un apposito menu; si possono disegnare cerchi, quadrati, linee ed ellissi ricorrendo a funzioni speciali; si può fare uso di una griglia che consente di tracciare linee perfettamente diritte e di allineare figure diverse; potete anche ingrandire alcune sezioni dell'immagine per definirne i dettagli; infine, è disponibile una varietà di fonti per chi vuole aggiungere del testo alle immagini.

Deluxe Paint ha però alcune caratteristiche che lo distinguono dagli altri programmi. Innanzitutto, può produrre immagini in tre risoluzioni diverse – 320 per 200, 640 per 200 e 640 per 400 pixel – (Deluxe Paint II permette l'uso con qualunque risoluzione e formato) e inoltre consente all'utente di definire e modificare un proprio pennello cromatico. Tutto ciò che dise-

gnate sullo schermo di Deluxe Paint può essere selezionato con il menu brush per essere poi usato come pennello custom (gestito dall'utente). Una volta selezionato, il pennello custom può essere utilizzato per disegnare dei solidi, tracciare linee a mano libera o fare una qualsiasi delle cose consentite dai pennelli standard. In più è possibile modificare il pennello, cambiandone le dimensioni, la forma e il colore, oppure capovolgendolo, ruotandolo o deformandolo attraverso i comandi flip, rotate e bend. Una volta ottenuto il pennello desiderato, l'utente può utilizzarlo in combinazione con diversi modi brush per aggiungere dettagli particolari o per modificare i colori dello schermo ottenendo effetti più o meno raffinati, a seconda delle esigenze specifiche. I modi shade, blend e smear permettono rispettivamente di sfumare, fondere e sbavare i colori (come se fossero freschi), mentre il modo cycle fa comparire a turno tutti i colori presenti nella palette. Saper usare efficacemente i pennelli custom è fondamentale se si vuole ottenere il massimo da Deluxe Paint. Prima di iniziare a lavorare con Deluxe Paint è opportuno creare un disco di lavoro sul quale immagazzinare le immagini da voi create.

COME CREARE UN DISCO DI LAVORO

È preferibile non utilizzare il disco di Deluxe Paint per memorizzare le figure di vostra ideazione, anche per il semplice motivo che il disco è praticamente pieno. Inoltre, è consigliabile posizionare l'apposita linguetta presente sul disco sulla posizione di protezione-scrittura, per evitare di cancellare accidentalmente qualche parte fondamentale del programma rovinando la copia di Deluxe Paint. Con un dischetto nuovo l'utente avrà spazio a volontà per le sue creazioni e non rischierà di rovinare il disco originale. Per inizializzare il disco di lavoro seguite attentamente la seguente procedura:

1. Accendete l'Amiga e caricate il Workbench dall'apposito dischetto.
2. Se avete un solo disk drive, togliete il disco del Workbench e inserite un dischetto nuovo (o un disco che non vi serve più). Se avete due drive, inserite il nuovo disco nel secondo drive.
3. Quando appare sullo schermo del Workbench l'icona raffigurante il disco vuoto, selezionate con la freccetta controllata dal mouse l'icona e, successivamente, il comando **Initialize** del menu **Disk**; seguite quindi le istruzioni per l'inizializzazione che appaiono sullo schermo.
4. Quando il disco è stato inizializzato, puntate col mouse la seconda icona del disco e premete due volte il bottone per aprirla e guardarne il contenuto. Apparirà un cestino (trashcan).

5. Aprite l'icona del disco Workbench per controllarne il contenuto. Una volta aperta, spostate la figura del cassetto contrassegnato come **Empty** dalla finestra del Workbench alla finestra del vostro disco, in modo da copiare il cassetto vuoto sul vostro dischetto.
6. Una volta copiato il cassetto vuoto, chiudete la finestra del Workbench, quindi selezionate l'icona del cassetto vuoto sul vostro disco e impartite il comando **Duplicate** presente nel menu **Workbench** per duplicare il cassetto vuoto. Ripetete questa procedura per altre due volte, in modo da ottenere tre copie del cassetto vuoto.
7. A questo punto dovrebbero esserci quattro icone del cassetto nella finestra del vostro disco. È possibile che le icone siano sovrapposte l'una all'altra e che quindi dobbiate spostarle per vederle tutte. Per dare un nome a ogni cassetto, selezionate il comando **Rename** dal menu **Workbench**. Quando appare la scritta al centro dello schermo, selezionatela puntandola col mouse e premendo il bottone, premete più volte il tasto DEL per cancellarne il contenuto, digitate un nuovo nome e premete RETURN quando avete terminato (usate i tasti cursore o BACKSPACE per correggere eventuali errori). Chiamate un cassetto **lo-res**, uno **med-res**, uno **hi-res** e l'ultimo **brush**.
8. Selezionate l'icona del vostro disco e il comando **Rename**, quindi ripetete le operazioni del punto 7 per dare un nome al vostro disco.

A questo punto avete un disco per lavorare con Deluxe Paint. Quando memorizzerete immagini e pennelli su questo disco, Deluxe Paint utilizzerà automaticamente i cassetti da voi appena creati. Tutte le immagini a bassa risoluzione vengono immagazzinate nel cassetto **lo-res**, quelle a risoluzione media nel cassetto **med-res** e quelle ad alta risoluzione nel cassetto **hi-res**. I pennelli vengono invece conservati nel cassetto **brush**.

Per evitare di dover creare un altro disco seguendo questa procedura noiosa, è sufficiente tenere il disco appena creato come copia standard di disco di lavoro vuoto. Tutte le volte che l'utente avrà bisogno di un nuovo disco non dovrà fare altro che copiare questo disco modello seguendo le istruzioni riportate nel manuale *Introduzione all'Amiga*.

SELEZIONE DELLA RISOLUZIONE E DEI COLORI DELLO SCHERMO

Quando avviate l'Amiga col Kickstart (operazione necessaria solo con il modello 1000) e poi inserite il disco di Deluxe Paint, immettete il comando **dpaint**, dopo il segno **1>**, e premete RETURN (Deluxe Paint II parte da solo oppure si seleziona l'icona). In questo modo, l'Amiga carica Deluxe Paint, partendo

con una risoluzione di 320 per 200 pixel a 32 colori. Se l'utente desidera creare immagini che utilizzino schermi di diversa risoluzione, è sufficiente caricare Deluxe Paint con un comando diverso. Immettendo **dpaint med** viene caricata una versione di Deluxe Paint che utilizza uno schermo a risoluzione 640 per 200; digitando **dpaint hi** si carica una versione con risoluzione 640 per 400 (in Deluxe Paint II la risoluzione e i colori si scelgono tramite un apposito requester). Sia nella media che nella alta risoluzione, il numero massimo di colori disponibili è 16.

L'utente può anche scegliere il numero di colori di cui disporre in seguito, quando carica il programma, semplicemente scegliendo il numero di bitplane che vuole utilizzare per i disegni. Tratteremo più ampiamente l'argomento dei bitplane nel quarto capitolo. Per ora basti sapere che il numero di bitplane determina il numero dei colori utilizzabili; di conseguenza, più bitplane utilizzate, più spazio occupate nella RAM. Cinque bitplane corrispondono a 32 colori, quattro bitplane a 16 colori, tre bitplane a 8 colori, due bitplane a 4 colori e un solo bitplane a 2 colori. Per inserire il numero di bitplane desiderato è sufficiente scrivere la relativa cifra dopo il comando **dpaint**. Ad esempio, per ottenere una versione di Deluxe Paint con schermo ad alta risoluzione e tre bitplane (8 colori), dovete immettere il comando **dpaint hi 3**. Per uno schermo a bassa risoluzione e due bitplane (4 colori), il comando è **dpaint lo 2**.

Ma perché mai si devono usare diverse risoluzioni e diverse quantità di bitplane? Il motivo principale è che l'utente ha la possibilità di salvare su disco le immagini realizzate per utilizzarle in seguito con altri programmi. Il problema è che alcuni di questi programmi richiedono che l'immagine rispetti una risoluzione specifica e un numero di bitplane fisso. Ad esempio, Deluxe Video (programma di animazione trattato nell'undicesimo capitolo) accetta solo immagini a bassa risoluzione con tre bitplane.

Una seconda ragione per cui è conveniente poter scegliere fra diversi tipi di risoluzione è che ognuno di essi offre vantaggi specifici. Per colorare un'immagine a bassa risoluzione l'utente può utilizzare 32 tinte. Una vasta gamma cromatica applicata a una figura a bassa risoluzione rende a volte la figura stessa più dettagliata e realistica di un'immagine analoga realizzata con uno schermo ad alta risoluzione, ma con meno colori. Le immagini ad alta risoluzione hanno invece il vantaggio di presentare linee curve e diagonali definite con maggior precisione, e offrono all'utente la possibilità di inserire tratteggi più raffinati, grazie alla sottigliezza delle linee e alla risoluzione superiore. Lo schermo ad alta risoluzione presenta un leggero tremolio (il motivo è spiegato nel prossimo capitolo), ma quando si riproducono le immagini su carta o con una macchina fotografica il tremolio non ha alcun effetto.

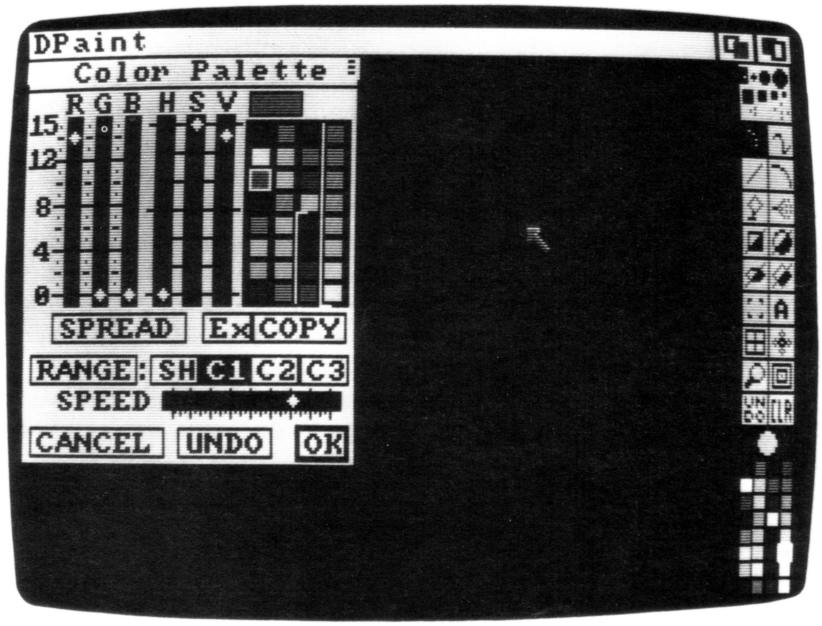
IMPOSTAZIONE DI UNA PALETTE

Dopo aver caricato Deluxe Paint e prima di iniziare a disegnare, è opportuno considerare in tutta calma quali colori si desidera utilizzare. Deluxe Paint dispone di un proprio set di colori prefissati, ma l'utente potrebbe desiderare, per l'immagine da lui creata, una palette (cioè una tavolozza) di colori diversi. Ad esempio, se dovete disegnare una foresta, una vasta gamma di soli verdi e marroni risulterà molto più utile dei rosa e dei rossi compresi nella palette preselezionata.

Per cambiare i colori della palette, aprite la finestra corrispondente premendo il tasto relativo alla lettera **p**. La finestra della palette appare come nella Figura 3-1. Ogni singolo colore incluso nella palette può essere modificato selezionandolo col puntatore del mouse e regolando i cursori presenti sulla sinistra. Ci sono due gruppi di tre cursori ciascuno. Il primo gruppo, contrassegnato dalle lettere **RGB**, definisce le percentuali di rosso, verde e blu contenute nel colore selezionato; il secondo gruppo, contrassegnato dalle lettere **HSV**, fissa la tonalità, la saturazione e il valore del colore selezionato. Valore è il termine utilizzato in Deluxe Paint per indicare l'intensità di un colore.

Figura 3-1.

La finestra della palette.



Quando l'utente modifica un colore usando uno dei due set di cursori, il programma adatta automaticamente l'altro set di cursori alle modifiche appena apportate. Sta a voi scegliere quale dei due metodi utilizzare. Il metodo **RGB** è più semplice se

si pensa in termini di combinazioni dei colori fondamentali; il metodo **HSV** è più semplice se si sceglie una tonalità di base e poi la si schiarisce o scurisce a piacimento con i cursori di saturazione e valore. Se non sapete molto della creazione dei colori coi metodi RGB e HSV (spesso indicato con HIS), sarebbe opportuno leggerne i principi, esposti nel secondo capitolo.

I cinque box contrassegnati dai nomi **RANGE**, **SH**, **C1**, **C2** e **C3** controllano le quattro gamme di colori utilizzate da Deluxe Paint nel caso in cui l'utente usi speciali modi brush (ampiamente trattati nelle prossime pagine). Le quattro gamme cromatiche sono la shade (gamma delle ombre – **SH**) e le tre gamme dei cicli di colori: color-cycle 1 (**C1**), color-cycle 2 (**C2**) e color-cycle 3 (**C3**). Deluxe Paint lavora con la gamma **SH** quando l'utente seleziona i modi brush **Blend** e **Shade**, mentre fa ruotare i colori delle gamme **C1**, **C2** e **C3** quando l'operatore usa il modo brush **Cycle** o il comando **Cycle** incluso nel menu **Picture**. Tutta questa serie di colori è stata preselezionata e può essere visualizzata selezionando **SH**, **C1**, **C2** o **C3**. Sulla palette appare una parentesi bianca che mostra quali sono i colori inclusi nella serie richiesta e i punti in cui la serie inizia e finisce.

Le gamme prefissate possono essere modificate a seconda delle esigenze. Per cambiare una gamma di colori, selezionate innanzitutto la serie che volete modificare fra le quattro disponibili. Poi scegliete uno dei colori della palette come colore di partenza della nuova serie; selezionate quindi **RANGE.** Sullo schermo appare un puntatore **TO**. Usate questo indicatore per scegliere il colore finale della nuova gamma. Fatto questo, il puntatore sparisce e la gamma desiderata è pronta.

La gamma shade funziona particolarmente bene quando viene settata su una serie di gradazioni chiare e scure della stessa tonalità. Un buon esempio è dato dalla sequenza di grigi che occupa gli ultimi 12 riquadri della palette standard di Deluxe Paint. Quando l'utente utilizza i modi brush **Blend** e **Shade**, Deluxe Paint si muove lungo questa sequenza schiarendo e scurendo i colori sullo schermo.

Quando l'operatore chiede a Deluxe Paint una rotazione di colori, il programma fa ruotare i colori compresi nella gamma cycle selezionata, alla velocità fissata per quella gamma dal cursore **SPEED** posto immediatamente al di sotto dei regolatori delle gamme cromatiche. Per cambiare la velocità di un ciclo di colori bisogna prima selezionare il ciclo interessato e poi spostare il cursore **SPEED**.

Dal momento che vi sono tre diverse gamme cycle, l'utente può approfittarne per metterle una dentro l'altra allo scopo di ottenere effetti molto bizzarri quando la rotazione viene visualizzata sullo schermo. Se volete cancellare tutte le gamme cromatiche contenute nella palette, è sufficiente selezionare la gamma, scegliere un colore, selezionare **RANGE** e usare il punta-

tore **TO** per impostare ancora lo stesso colore, col risultato di creare una gamma monocromatica.

Quando è stata adattata alle proprie esigenze, la palette può essere definitivamente memorizzata col comando **OK**; in caso di ripensamenti, l'utente può tornare alla palette di partenza selezionando **CANCEL**, che elimina tutte le modifiche apportate. Per quante modifiche siano state fatte, non esiste la minima possibilità di cancellare la palette prefissata di Deluxe Paint. Quando ne avrete bisogno, potrete farla riapparire selezionando l'opzione **Default Palette** dal menu **Picture**.

COME REALIZZARE DISEGNI DI PRECISIONE

Usando gli strumenti grafici di Deluxe Paint, può capitare di dover sistemare linee, angoli retti, cerchi o altre figure in posizioni ben precise, in modo che non si sovrappongano ad altre figure già presenti sullo schermo. Si può sempre ricorrere all'ingrandimento per vedere bene ciò che si sta facendo, oppure si può sforzare la vista per cercare di esaminare ogni singolo pixel, ma esistono procedimenti più semplici per mettere tutto al posto giusto.

Posizionamento del pennello col comando Coordinates

Uno dei metodi più semplici per posizionare il pennello sullo schermo consiste nel selezionare il comando **Coordinates** nel menu **Prefs**. Appariranno, sul lato destro della linea d'intestazione (title bar), i numeri che indicano le coordinate del puntatore che muove il pennello. I numeri indicano la distanza in pixel dall'angolo superiore sinistro al puntatore; la prima cifra si riferisce al numero di pixel in orizzontale, la seconda a quelli in verticale. Annotando le coordinate visualizzate quando avete finito di tracciare una linea o l'angolo di un quadrilatero, potrete servirvene per iniziare una nuova figura nello stesso punto. Supponete, ad esempio, di voler tracciare una serie di linee che partono da uno stesso punto come raggi. Partendo sempre dalle coordinate della prima linea, sarete sicuri di tracciare linee con l'origine in comune.

Il sistema delle coordinate funziona in maniera diversa quando si preme il tasto del mouse e si muove il cursore sullo schermo. Invece di dare la posizione del cursore in termini di distanza in pixel dall'angolo superiore sinistro, viene indicata la distanza del cursore dal punto in cui il tasto del mouse è stato premuto per la prima volta. Se state disegnando un oggetto, potete misurarne l'ampiezza. Ad esempio, se volete disegnare un rettangolo largo 10 pixel e alto 16, fissate il primo angolo e poi muovete il puntatore per fissare l'angolo opposto, seguendo la progressione dei numeri delle coordinate finché raggiungete le misure desiderate.

Come tracciare linee rette col tasto SHIFT

Il tasto SHIFT contribuisce ulteriormente alla precisione del cursore. Se tenete premuto il tasto SHIFT mentre muovete il mouse, Deluxe Paint sposterà il pennello in direzione solo orizzontale o solo verticale, a seconda di come muovete il mouse. Per muovere il pennello in altre direzioni dovete lasciare il tasto SHIFT.

Questa possibilità è molto utile per disegnare linee perfettamente diritte o per muovere un pennello sullo schermo seguendo una direzione precisa. Ad esempio, supponete di dover tracciare una linea perfettamente retta che attraversi tutto lo schermo. Selezionate l'opzione per tracciare le linee e poi spostate l'indicatore del mouse nel punto in cui volete che la linea inizi. Prima di premere il tasto sinistro del mouse per iniziare a tracciare la retta, premete il tasto SHIFT e tenetelo abbassato. Ora schiacciate il tasto sinistro cominciando a muovere il mouse di lato. Mentre voi eseguite il movimento, il tasto SHIFT mantiene la linea lungo una traccia perfettamente orizzontale, anche se il mouse non segue una direzione esattamente orizzontale.

Allineamento delle immagini con la griglia

La griglia (grid) si rivela ancora più utile dei procedimenti illustrati in precedenza per posizionare linee ed oggetti. Quando selezionate l'opzione grid sul pannello di controllo, viene costruita sullo schermo una griglia di linee invisibili e i movimenti del cursore, anche durante l'uso di altre funzioni, sono limitati alle intersezioni delle linee tracciate dalla griglia. Quando l'utente impiega questa funzione per il disegno a mano libera con un pennello, la griglia limita il tracciato del pennello alle proprie intersezioni. Quando invece l'utente traccia linee rette e curve, la griglia ritocca gli estremi delle linee tracciate (che devono sempre combaciare con una delle sue intersezioni). Allo stesso modo vengono elaborati gli angoli di un rettangolo nonché il centro e i bordi di figure circolari ed ellittiche.

Se azionate la griglia e iniziate a disegnare, risulta molto facile allineare le figure realizzate. Ad esempio, se volete disegnare un quadrilatero sovrastato da una cupola, azionate la griglia e tracciate il rettangolo desiderato. I suoi angoli saranno necessariamente limitati alle intersezioni del reticolo invisibile, quindi il cursore, muovendosi sullo schermo, passerà da un'intersezione all'altra. Una volta creato il rettangolo, selezionate l'opzione per tracciare linee curve.

Dato che la griglia limita i movimenti del cursore alle intersezioni, non sarà difficile posizionare il puntatore su uno dei due angoli superiori del quadrilatero; è sufficiente posizionare il centro della croce su un angolo. Premete il tasto sinistro del mouse e muovetevi lungo il lato superiore del rettangolo, dopodiché rilasciate il tasto. Gli estremi della curva corrisponderanno perfettamente agli angoli del rettangolo. A questo punto muovete il mouse per imprimere alla linea la curvatura desiderata.

Se le intersezioni prefissate della griglia (una ogni 8 pixel) sono troppo fitte o troppo rade per i vostri scopi, potete modificarle selezionando l'opzione apposita (grid) con il bottone destro del mouse. Appare sullo schermo una sezione della griglia, definita "griglia di dimensionamento" (sizing grid). Le coordinate che appaiono sul lato destro della linea del menu indicano la posizione dell'angolo superiore sinistro della griglia di dimensionamento. Tenendo premuto il tasto sinistro e muovendo il mouse è possibile modificare le dimensioni della griglia. Mentre fate scorrere il mouse, i numeri delle coordinate cambiano, indicando altezza e larghezza in pixel di ciascun quadrato della griglia. Una volta ottenuta la grandezza desiderata, si può rilasciare il tasto del mouse. La griglia sparisce e tutte le seguenti applicazioni grafiche che faranno uso della griglia terranno conto della nuova dislocazione delle intersezioni.

La griglia preselezionata è allineata con i limiti superiore e sinistro dello schermo, ma, volendo, può essere riposizionata. Innanzitutto, selezionate l'opzione grid col tasto destro del mouse. Riapparirà sullo schermo la griglia di dimensionamento, ma questa volta dovrete posizionarla sullo schermo per determinare la locazione delle linee del reticolo, e non la distanza fra una linea e l'altra. Piazzate le intersezioni della griglia di dimensionamento nella posizione desiderata, premete il bottone sinistro del mouse e l'intera griglia sarà riallineata secondo le modifiche apportate alla griglia di dimensionamento.

Quando usate la griglia per l'inserimento di testi, essa limita la libertà di movimento del cursore testo. Quando iniziate a scrivere, le lettere non sono sotto l'influenza della griglia, ma se decidete di trasferire il testo in un'altra sezione dello schermo, la sua nuova posizione dovrà rispettare il reticolo della griglia. Stabilendo una griglia molto rada è possibile usare le intersezioni come stop di tabulazione o interlinee. È sufficiente scrivere quello che si desidera e riposizionare il testo col puntatore. La griglia facilita l'incolonnamento e l'allineamento dei testi.

Una delle applicazioni più interessanti della griglia è la creazione di modelli grafici di sfondo. Per provare, caricate innanzitutto **Patterns** dal disco di Deluxe Paint. Nell'angolo superiore destro dell'immagine visualizzata sono riportate otto

diverse matrici, contenute in box gialli. L'utente può selezionare uno qualsiasi di questi motivi come pennello custom, per riprodurlo poi sull'intera superficie dello schermo, utilizzando la griglia per allineare il tutto.

Per sperimentare questa funzione, iniziate utilizzando il comando di selezione dei pennelli per scegliere il box contenente il muro di mattoni; selezionate l'intero contenuto del riquadro, facendo però attenzione a non includere parti di contorno giallo. Osservando le coordinate indicate al momento della creazione del pennello, si nota che il riquadro misura esattamente 19 pixel di base e 11 di altezza. Premete il tasto **j** per passare all'altro schermo grafico, sul quale potete disegnare liberamente senza pericolo di coprire le altre matrici grafiche. A questo punto dimensionate nuovamente la griglia in modo che sia un pixel più alta e più larga del vostro pennello; dovrà misurare quindi 20 pixel per 12. Una volta definita la griglia, azionatela, assicuratevi che il pennello sia predisposto per il disegno a mano libera e iniziate finalmente a dipingere. A questo punto lo schermo comincerà a riempirsi di blocchi di mattoni perfettamente allineati.

Una volta completato lo schermo, provate a disegnare qualcosa sul "muro". Se fate qualcosa che non vi piace e non riuscite a cancellarlo usando semplicemente **UNDO**, non vi sarà di grande utilità nemmeno il colore di sfondo, perché la mattonata riprodotta non è formata da un unico colore. Azionate la griglia, se non l'avete già fatto, e selezionate una sezione qualsiasi del muro come pennello, ricominciando a dipingere nei punti che avete pasticciato coi vostri tentativi. Se nel frattempo la griglia non ha subito modificazioni, il pennello cancellerà gli scarabocchi coprendoli con nuovi blocchi di mattoni perfettamente allineati con quelli disegnati precedentemente.

I MODI BRUSH

La creazione di pennelli custom, ricavati da una porzione di immagine già visualizzata sullo schermo, e l'uso dei comandi del menu **Brush** per capovolgere, far girare, ricolorare e ristrutturare i pennelli finché si perviene alla forma e al colore desiderati, costituiscono uno tra gli aspetti più divertenti dell'uso di Deluxe Paint. Una volta ricavato un pennello custom, oppure un semplice pennello selezionato dal pannello di controllo, il modo brush che selezionate dal menu **Mode** determina le modalità secondo cui il pennello scelto disegna sullo schermo. I modi **Object**, **Color** e **Replace** forniscono le opzioni per disegnare con un pennello custom. Se si desidera utilizzare per l'immagine uno sfondo particolare, è possibile ricorrere ad altri tre modi brush che modificano i colori già presenti sullo schermo: **Smear**, **Shade** e **Blend**. **Cycle**, un altro modo brush, serve a

creare effetti speciali. I paragrafi che seguono descrivono dettagliatamente ciascuno dei vari modi brush. Attraverso la lettura di un paragrafo dimostrativo incluso in questo capitolo verificherete inoltre il funzionamento di tutti i modi brush.

Il modo Object

Ogni volta che si ricorre al comando di selezione dei pennelli, per visualizzare un rettangolo sullo schermo e copiarne il contenuto come pennello, Deluxe Paint utilizza automaticamente il modo **Object**. Col modo **Object** si ha l'impressione che Deluxe Paint possa vedere qual è il colore di sfondo. Qualsiasi pixel che all'interno del rettangolo abbia lo stesso colore dello sfondo diventa trasparente nel pennello. In altre parole, il rettangolo del pennello custom esclude tutti i pixel dello sfondo. Quando si colora con un pennello in modo **Object**, vengono usati tutti i colori del pennello, utilizzando la forma dell'oggetto selezionato come pennello, ma senza i colori dello sfondo.

Il modo Color

Una volta creato un pennello custom – ad esempio, se si disegna un arcobaleno con uno sfondo bianco, si ricorre poi al comando di selezione dei pennelli per selezionarlo come pennello custom – Deluxe Paint nel modo **Object** utilizza automaticamente la forma incurvata dell'arcobaleno, con tutti i suoi colori, come pennello custom. In seguito, se si sceglie il modo **Color**, Deluxe Paint trasforma il pennello selezionato in una sagoma, colorandola di un unico colore: il colore di primo piano presente in quel momento. Se, nel nostro esempio, il colore di primo piano è il verde, Deluxe Paint utilizza la figura dell'arcobaleno colorandola però unicamente di verde.

Il modo Replace

Dopo aver selezionato un pennello custom, è possibile disporre anche del modo **Replace**. Con esso, Deluxe Paint include nel pennello il colore dello sfondo che prima è stato considerato trasparente. Nel pennello custom dell'ultimo esempio verrebbero mostrati tutti i colori dell'arcobaleno, compreso il bianco dello sfondo utilizzato nel rettangolo al momento della selezione del pennello. Quando si dipinge con un pennello custom nel modo **Replace**, si dipinge con un intero rettangolo, perché il colore dello sfondo viene usato allo stesso modo degli altri colori del pennello.

Il modo Smear

Utilizzare il modo **Smear** di Deluxe Paint equivale, in termini pratici, a mischiare diversi colori a olio usando le dita. Quando si colora in modo **Smear**, il pennello rimescola i pixel già presenti sullo schermo, mischiandone le tinte. È possibile usare sia pennelli standard che pennelli custom; nel secondo caso, Deluxe Paint usa la stessa forma di pennello che utilizzerebbe nel modo **Color**.

Il modo Shade

Nel modo **Shade** il pennello interessa solo quei pixel il cui colore è incluso nella gamma di sfumature della palette. Tutti i colori che non sono inclusi in questa gamma vengono lasciati intatti. Si può usare sia un pennello standard, sia un pennello custom; quest'ultimo utilizza solo il profilo dell'oggetto.

Se ci si avvale del modo **Shade**, tutte le volte che si tiene premuto il tasto sinistro del mouse e si fa passare il pennello su pixel dello stesso colore di quelli inclusi nella gamma di sfumature della palette, i pixel toccati dal pennello cambiano colore, assumendo la tinta immediatamente inferiore nella scala delle sfumature. Premendo il tasto destro i pixel si modificano, invece, nella tinta immediatamente superiore.

Qualora venga creata nella finestra della palette una gamma di sfumature in progressione dallo scuro al chiaro, è possibile servirsi del tasto sinistro del mouse per schiarire, e del tasto destro per scurire, i pixel ripassati col pennello. I pixel non muteranno colore assumendo tonalità non incluse nella gamma, cosicché sarà impossibile schiarirli o scurirli oltre le due sfumature poste agli estremi opposti della gamma selezionata.

Il modo Blend

Il modo **Blend** è in parte simile al modo **Smear**; quando si muove il pennello nel modo **Blend** toccando due diversi colori, entrambi inclusi nella gamma di sfumature, il pennello mescola parzialmente i due colori. La differenza consiste nel fatto che nel modo **Blend** il pennello non si limita a spostare i pixel di un colore in un'area di un altro colore: il pennello confronta i colori che sta muovendo con quelli su cui sta passando e cerca una tonalità intermedia fra i due all'interno della scala di sfumature. Se i due colori, all'interno della gamma, sono vicini, Deluxe Paint utilizza il colore dei pixel da cui il pennello è partito. Operando con una gamma di sfumature che varia progressivamente dallo scuro al chiaro, l'utente ha la possibilità di usare il modo **Blend** allo scopo di offuscare la linea che separa due diversi colori, servendosi di tutte le sfumature di colore intermedie al fine di conseguire un passaggio armonioso e graduale da una tinta all'altra. Seguendo l'esempio illustrato più avanti, comprenderete a pieno questa tecnica.

Il modo Cycle

Il modo **Cycle** è ottimo per creare interessanti effetti speciali. Quando si colora in modo **Cycle**, il pennello fa ruotare alternativamente tutti i colori selezionati all'interno di una delle gamme cycle, lasciando dietro di sé una scia multicolore. Per scegliere una qualsiasi delle tre serie cycle (**C1**, **C2**, **C3**), si deve selezionare uno qualunque dei colori del pannello di controllo che appartiene alla serie cycle desiderata. L'operatore può quindi iniziare a muovere il pennello. È possibile regolare la velocità di rotazione dei colori servendosi del cursore **SPEED** che, all'interno della finestra della palette, corrisponde a quella gamma cromatica.

Se si usa il modo **Cycle** per visualizzare quadrilateri, cerchi, linee o altre forme create attraverso i comandi del pannello di controllo di Deluxe Paint, ogni oggetto creato è dipinto con un unico colore, ma ogni figura successiva è colorata con una tinta diversa compresa nella serie cycle.

Dopo aver dipinto le immagini di uno schermo con i colori della serie cycle, è possibile, per ogni colore del video compreso nella gamma cycle selezionata, ottenere una completa rotazione cromatica, semplicemente premendo il tasto TAB.

APPLICAZIONI DIMOSTRATIVE DEI DIVERSI MODI BRUSH

Ora che conoscete le funzioni dei diversi modi brush, potete cimentarvi in qualche applicazione pratica che si serve dei modi brush per ottenere determinati effetti. Tutti gli esempi sono piuttosto semplici, tuttavia alcuni di essi consentono di conseguire risultati molto particolari, che potrebbero fornire dei suggerimenti per le immagini di vostra ideazione. Gli esempi utilizzano il modo lo-res (bassa risoluzione) di Deluxe Paint. Pertanto, prima di iniziare, dovete avviare il programma digitando **dpaint**, quando appare l'indicazione **1>**.

Come usare i modi Object e Color per delineare una figura

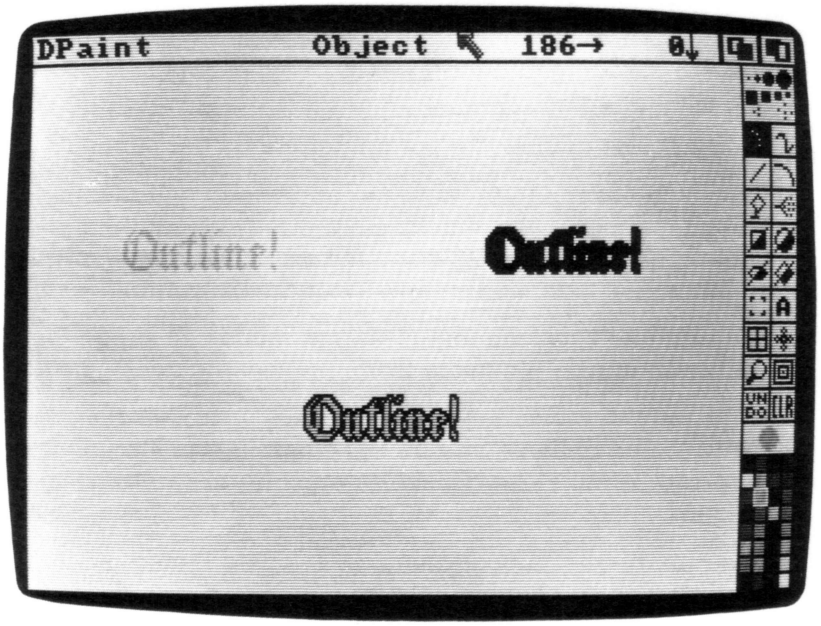
Quando create un oggetto usando colori che non producono un forte contrasto con il colore di sfondo dello schermo, è probabile che vogliate far risaltare meglio l'oggetto in questione con un colore contrastante con lo sfondo. Ad esempio, un oggetto arancione su sfondo bianco risalterà più nettamente se i suoi contorni saranno stati tracciati in nero. È sempre possibile delineare chiaramente un oggetto tracciandone con cura i contorni, ma ciò comporta una notevole perdita di tempo, se l'oggetto è di forma complessa. Selezionando l'oggetto come pennello e utilizzando i modi brush **Object** e **Color**, è

possibile contornare l'oggetto nel giro di pochi secondi, indipendentemente dalla complessità della figura. Verificate di persona seguendo queste istruzioni:

1. Usate il tasto destro del mouse per selezionare il bianco come colore di sfondo e poi immettete **CLR** dal pannello di controllo per liberare lo schermo e colorarlo interamente di bianco.
2. Selezionate **Load Fonts** dal menu **Font** e scegliete una fonte piuttosto grande, ad esempio la **sapphire-19** (zaffiro-19).
3. Selezionate l'opzione text (testo) in modo che sullo schermo appaia il cursore testo; posizionate quest'ultimo premendo il bottone del mouse.
4. Selezionate dalla palette un blu chiaro e digitate un messaggio sul video. La scritta apparirà in caratteri azzurri.
5. Terminata questa operazione, selezionate l'intera serie di caratteri come pennello custom.
6. Scegliete **Color** dal menu **Mode** e il colore nero dalla palette, in modo da dipingere il vostro pennello di nero.
7. Scegliete **Coordinates** dal menu **Prefs** per vedere le coordinate visualizzate nella linea d'intestazione.
8. Selezionate la funzione rettangolo vuoto (hollow rectangle, la quarta dall' angolo superiore sinistro).
9. Muovete il pennello in una sezione libera dello schermo, tenete premuto il pulsante sinistro del mouse e spostate il pennello in basso e verso destra finché le coordinate indicano una dimensione di 2 pixel per 2. Si otterrà come risultato una serie illeggibile di macchie nere, simili a quelle visualizzate nella parte destra dello schermo riportato nella Figura 3-2.
10. Selezionate **Object** dal menu **Mode** per restituire al vostro pennello il suo originario colore azzurro.
11. Posizionate il pennello al centro del "rettangolo" creato in precedenza, in modo da poter visualizzare il testo con un contorno scuro. Premete il tasto sinistro del mouse per copiare il pennello. Spostate il pennello e il gioco è fatto. Avete ricavato un testo azzurro contornato di nero, simile a quello che potete vedere nella parte inferiore dello schermo, nella Figura 3-2 (riportata nella pagina successiva).

Figura 3-2.

Il "rettangolo" sulla destra dello schermo è stato creato con un pennello testo. Nella parte inferiore dello schermo è visualizzato il pennello originale posto al centro del rettangolo in modo da creare una scritta contornata che risalti sullo sfondo.



12. Selezionate il testo delineato come nuovo pennello custom e memorizzatelo su dischetto. Lo utilizzerete in un esempio successivo.

In questo esempio, quando avete utilizzato il pennello custom nel modo **Object**, Deluxe Paint si è sempre servito dei colori originari (in questo caso l'azzurro) contenuti nel pennello quando questo è stato selezionato come pennello custom per la prima volta. Una volta convertito il pennello custom nel modo **Color**, potete scegliere qualsiasi colore vi aggradi (in questo esempio, nella fase 6, potevate scegliere qualsiasi colore per far risaltare la scritta). Non appena tornate nel modo **Object**, il pennello riacquista i suoi colori originari.

Creazione di uno sfondo "sabbioso" col modo Smear

La maggior parte delle funzioni del pannello di controllo crea oggetti di un unico colore. La creazione, attraverso comandi standard, di figure che associno diversi colori in uno spazio ridotto, comporta una notevole perdita di tempo. Pensate, ad esempio, di dover disegnare dei palloni colorati su una spiaggia. Disegnare ogni singolo granello di sabbia della

spiaggia, alternando colori chiari, scuri e intermedi, può rivelarsi veramente noioso. È possibile, invece, usare un pennello nel modo Smear per ottenere i risultati desiderati senza sprecare tempo e fatica.

Nell'esempio che segue assicuratevi di usare la palette preselezionata, impostata sulla gamma di sfumature in modo da includere gli ultimi 12 colori della palette, che variano dal grigio scuro al grigio chiaro. Sebbene questi colori non siano necessari nel nostro esempio, potrete utilizzare i risultati in un esempio successivo che richieda l'uso della suddetta gamma di sfumature.

1. Cancellate lo schermo.
2. Scegliete un pennello circolare col pulsante destro del mouse e dimensionatelo a 10 pixel per 10.
3. Visualizzate sullo schermo tutti i 12 colori della gamma di sfumature in piccole macchie che si tocchino l'un l'altra e siano disposte a cerchio, come illustra la parte sinistra della Figura 3-3 (riportata nella pagina successiva).
4. Selezionate un pennello rotondo di piccole dimensioni.
5. Selezionate **Smear** dal menu **Mode**, muovete il pennello al centro delle macchie di colore, tenete premuto il tasto sinistro del mouse e incominciate a "pasticciare". Mescolate i colori con un movimento circolare partendo dal centro del cerchio di macchie, in modo da formare nel centro un miscuglio di piccole quantità di ogni colore. Il risultato ottenuto dovrebbe essere simile a ciò che si vede a destra nella Figura 3-3.
6. Una volta ottenuto un miscuglio di colori ben assortiti, incominciate a pasticciare partendo dal centro verso i bordi; grazie a questa operazione è possibile rendere il miscuglio più uniforme e più esteso, espandendolo verso l'esterno.
7. Quando avete prodotto una buona sezione di sfondo, selezionatene il centro come pennello custom, poi colorate l'intero schermo col pennello creato, ottenendo così lo sfondo sabbioso adatto. Si otterrano risultati simili a quelli della Figura 3-4.
8. Salvate la vostra creazione su disco selezionando il comando **Save As...** dal menu **Picture**. Utilizzerete questa immagine in un esempio successivo.

Figura 3-3.

I colori a sinistra sono le macchie così come appaiono prima di essere pasticciate. A destra sono invece riprodotti i colori dopo il trattamento nel modo Smear.

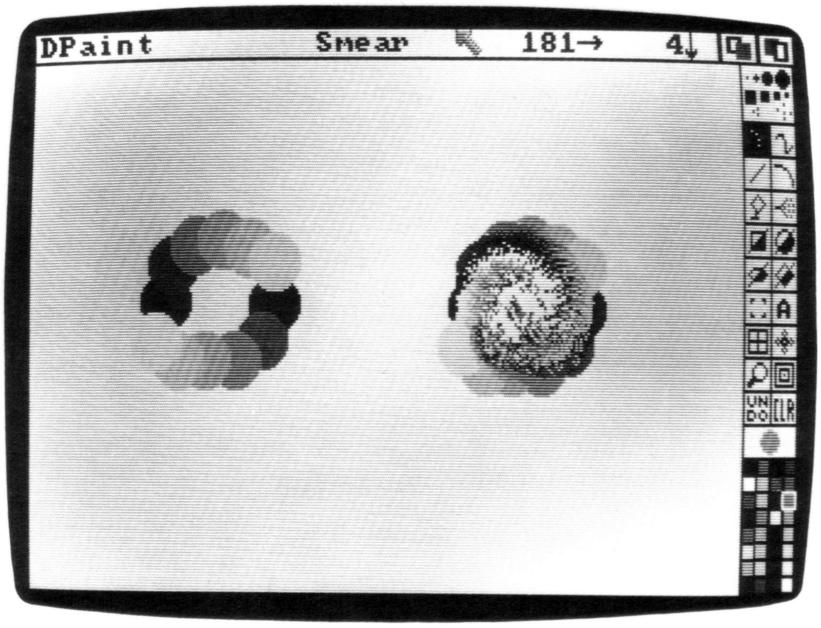
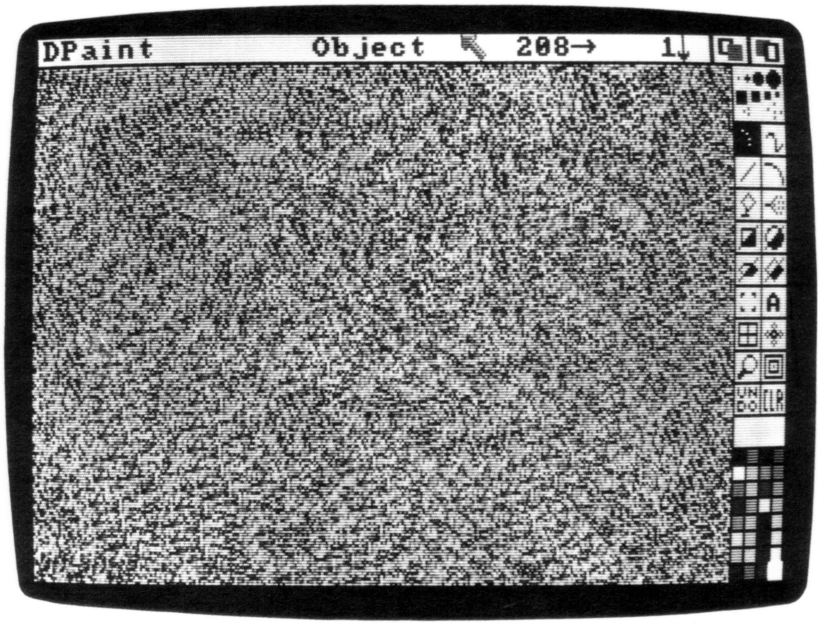


Figura 3-4.

Uno schermo pieno di sabbia.



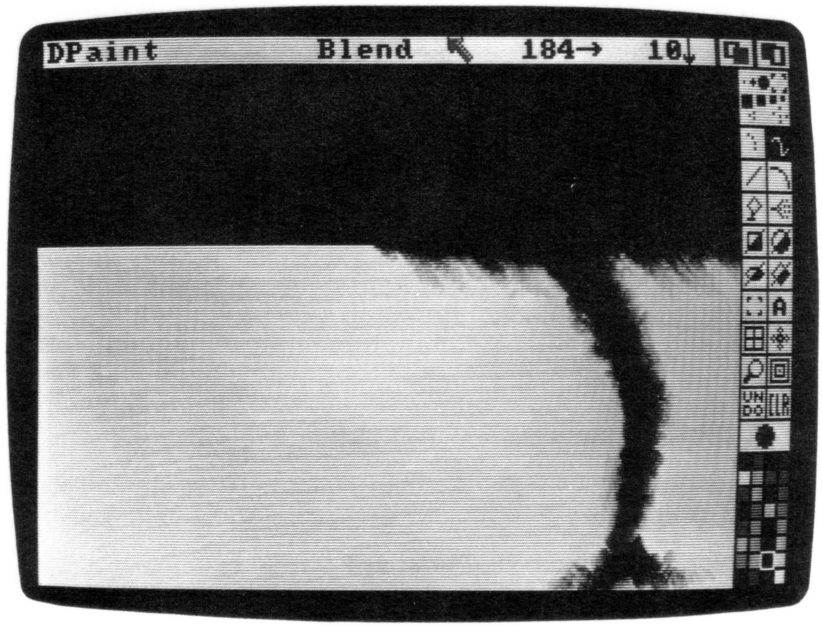
Creazione di nuvole di pioggia col modo Blend

Se avete osservato le nuvole, avrete senz'altro notato che presentano diverse tonalità di grigio, che variano dal grigio chiaro al grigio molto scuro, come nel caso di nuvole cariche di pioggia. Qualora si desiderassero creare con Deluxe Paint nuvole perfettamente sfumate, è consigliabile servirsi di un pennello nel modo **Blend**. Con questo sistema si può ottenere una vasta gamma di grigi partendo da due sole tonalità originarie.

1. Cancellate lo schermo (lo sfondo deve essere bianco).
2. Aprite la finestra della palette e assicuratevi che le 12 sfumature di grigio siano gli unici 12 colori inclusi nella gamma di sfumature, dopodiché chiudete la palette.
3. Riempite un terzo dello schermo, nella parte superiore, col grigio più scuro incluso nella serie delle sfumature.
4. Colorate i restanti due terzi utilizzando la sfumatura più chiara disponibile. Il risultato deve essere simile alla Figura 3-5 (riportata nella pagina successiva) nella parte sinistra.
5. Scegliete il pennello circolare più grande nel pannello di controllo, cioè quello all'estrema destra del video, e selezionate **Blend** dal menu **Mode**.
6. Portate il pennello nel settore superiore scuro dello schermo, tenete premuto il pulsante sinistro del mouse e spostate lentamente il pennello verso il basso e verso destra, finché entra nel settore chiaro del quadro. Il settore scuro dovrebbe sfumare in quello chiaro. Ripetete l'operazione più volte, passando da una sfumatura di grigio all'altra per creare una sorta di "effetto pioggia".
7. Per aggiungere una tromba d'aria portata dalle nuvole, è sufficiente continuare a prelevare colore dal settore scuro e riportarlo in basso, formando un'unica striscia, che si dovrebbe avvicinare progressivamente al fondo dello schermo a ogni vostra "pennellata". Continuate a muovere il pennello dall'alto verso il basso, imprimendo alla striscia una certa curvatura che dia l'impressione di una nuvola a forma d'imbuto. Alla fine dovreste ottenere un'immagine simile alla metà destra della Figura 3-5.

Figura 3-5.

Le due tonalità di grigio sulla sinistra dello schermo vengono fuse utilizzando un pennello nel modo Blend, per creare le nuvole e la tromba d'aria riprodotte sulla destra.



Come usare il modo Shade per creare ombre

Il manuale di Deluxe Paint spiega come creare velocemente l'ombra di un oggetto. È sufficiente selezionare l'oggetto come pennello, selezionare **Color** dal menu **Mode**, scegliere il nero come colore e visualizzare una copia dell'oggetto-pennello nero sullo schermo. Successivamente bisogna scegliere **Object** dal menu **Mode** e disporre l'oggetto sopra e leggermente a lato della sua "ombra". È inoltre possibile utilizzare il modo **Shade** per ottenere un'ombra più realistica su uno sfondo complesso, purchè lo sfondo sia stato creato usando colori le cui tonalità variano dal chiaro allo scuro, all'interno di una gamma di sfumature simile a quella dell'esempio dello sfondo sabbioso. Con questa tecnica non si creano solo ombre completamente nere; le ombre sono, più precisamente e realisticamente, lo sfondo originale scurito, il che consente di vedere una vera ombra proiettata sul disegno di sfondo originale.

Nell'esempio seguente viene spiegato come creare un'ombra per la scritta precedentemente contornata di nero, il tutto sullo sfondo sabbioso che avete salvato in un altro esempio.

1. Caricate lo sfondo sabbioso creato in precedenza e salvatelo come immagine.

2. Caricate la scritta bordata di nero e salvatela come pennello.

3. Collocate questo pennello nella posizione in cui preferite visualizzare la scritta.

4. Selezionate **Shade** dal menu **Mode**. Il pennello apparirà sul video come una sfumatura più chiara della sabbia, naturalmente conservando la sua forma.

5. Per scurire la sabbia sotto il pennello premete sei volte il tasto destro del mouse, evitando di muovere il mouse stesso. A ogni pressione la sabbia si fa un po' più scura. Il pennello, una volta allontanato dall'area scurita, sembra ancora più scuro; questo perché il modo **Shade** fa sempre apparire l'area sottostante più chiara di una tonalità, per consentire all'utente di distinguere il pennello.

6. Selezionate **Object** dal menu **Mode**. Il pennello apparirà sotto forma della scritta originaria. Ponete la scritta sopra e leggermente a sinistra dell'ombra e premete il tasto sinistro del mouse per copiarla. Si dovrebbero ottenere risultati simili a quelli esposti nella Figura 3-6. Osservando l'ombra da distanza ravvicinata, è ancora possibile distinguere i granelli di sabbia, nonostante siano più scuri degli originali.

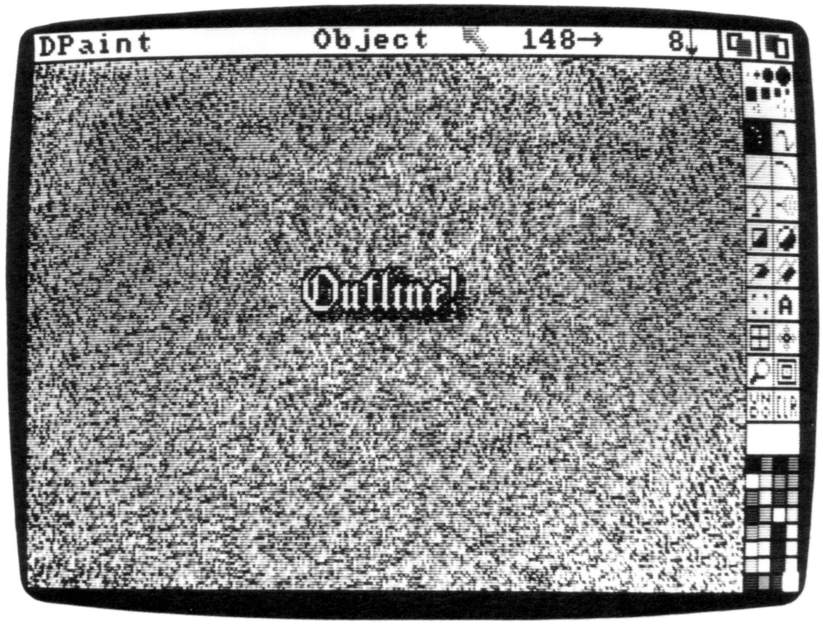


Figura 3-6.

Le lettere proiettano la loro ombra sulla sabbia, grazie all'uso di un pennello nel modo Shade.

Questa tecnica dell'ombra può essere utilizzata su qualsiasi sfondo disegnato nel modo **Shade**. Si possono usare sfondi di legno truciolato, griglie metalliche, campi erbosi o qualsiasi altra immagine utilizzi una gamma di colori della stessa tonalità suddivisa in più sfumature.

TRUCCHETTI UTILI

Utilizzando Deluxe Paint scoprirete molti trucchi ed espedienti per creare immagini con maggior semplicità. Ne riportiamo qui di seguito tre che potranno esservi molto utili.

Uso di abbreviazioni dei comandi

Alla fine del manuale di Deluxe Paint viene fornito un elenco di abbreviazioni, da impartirsi tramite tastiera, che consentono di sostituire i comandi del pannello di controllo e dei menu. Aprite il manuale a quella pagina, mettetelo bene in vista sul tavolo e, mentre lavorate, memorizzate con attenzione le lettere indicate. Ne vale sicuramente la pena. Scoprirete che il vostro lavoro può procedere molto più velocemente se, ad esempio, potete capovolgere un pennello sullo schermo premendo solo il tasto **x**, invece di dover aprire il menu **Brush**, selezionare **Flip** e poi **Horiz**.

Un ulteriore vantaggio offerto dall'uso della tastiera consiste nella possibilità di eliminare dal video sia il pannello di controllo, sia la riga d'intestazione, in modo da poter utilizzare l'intero schermo per le immagini. Inoltre, essendo necessaria solo la tastiera, non si dovrà continuamente riportare sullo schermo il pannello per cambiare i pennelli o per selezionare nuovi comandi e colori.

La maggior parte delle abbreviazioni per tastiera è facile da ricordare, trattandosi generalmente di lettere iniziali di alcune parole ben precise: **b**, ad esempio, indica la funzione brush. I tasti funzione collocati nella parte superiore della tastiera svolgono altre funzioni. Per poterli usare con maggiore semplicità, potete realizzare una striscia di carta che indichi le varie funzioni. Nell'Amiga 1000, immediatamente sopra i tasti funzione, è stata creata una rientranza apposita per contenere strisce di questo tipo. È sufficiente, quindi, ritagliare una striscia di carta o cartone della giusta misura e infilarla nella rientranza. Anche se le tastiere dell'A500 e dell'A2000 non offrono questa possibilità, lo spazio libero accanto ai tasti funzione è sufficiente ad accogliere una striscia di carta fissata con dello scotch.

Uso del monitor appoggiato su un fianco

Poiché la larghezza dello schermo di Deluxe Paint è superiore alla sua altezza, è facile cadere nella ripetitività e finire col realizzare sempre immagini più larghe che alte. Potete cambiare le proporzioni del vostro campo operativo sistemando il monitor verticalmente, appoggiato cioè su un fianco invece che sulla base. Sebbene ci voglia del tempo per abituarsi a tenere il mouse di lato mentre lo si muove e a selezionare i menu dal lato dello schermo invece che dalla parte superiore, scoprirete che operare su un video più alto che largo offre nuove prospettive. Se si stampano le immagini create con Deluxe Paint verticalmente su un foglio di carta (per i dettagli vedere la successiva sezione "Stampa delle immagini di Deluxe Paint"), si ha l'ulteriore vantaggio di poter vedere sullo schermo il foglio in tutta la sua lunghezza.

Copiatura su acetato di immagini tratte da riviste

Se non siete proprio un grande disegnatore a mano libera, o se lo siete ma non volete sprecare troppe energie, potete avvalervi dell'uso di fogli di acetato trasparenti (acquistabili presso qualunque negozio di cancelleria), di un pennarello con punta fine e di alcune riviste. Sovrapponendo l'acetato all'immagine di una rivista o di un libro, potete ricalcare l'immagine servendovi del pennarello. Terminata l'operazione, fissate l'acetato sullo schermo del monitor e seguite i tratti del pennarello col cursore di Deluxe Paint per riprodurre l'immagine sul video. Finché lavorerete da soli e a porte chiuse, nessuno verrà a conoscenza del vostro segreto.

STAMPA DELLE IMMAGINI DI DELUXE PAINT

Se possedete una stampante collegata all'Amiga in grado di riprodurre immagini, potete stampare qualsiasi creazione realizzata con Deluxe Paint, semplicemente selezionando il comando **Print** dal menu **Picture**. Deluxe Paint invia i dati relativi all'immagine al printer driver della stampante, precedentemente caricato con il software sistema all'inserimento del dischetto di Deluxe Paint. Il printer driver traduce i dati dell'immagine in un flusso di istruzioni che ne consentono la riproduzione.

Sebbene dall'interno di Deluxe Paint non vi sia alcuna possibilità di controllare il modo in cui l'immagine viene stampata, l'utente può realizzare diverse versioni stampate della stessa immagine modificando diversi parametri del driver attraverso il programma Preferences.

COME USARE IL PROGRAMMA PREFERENCES

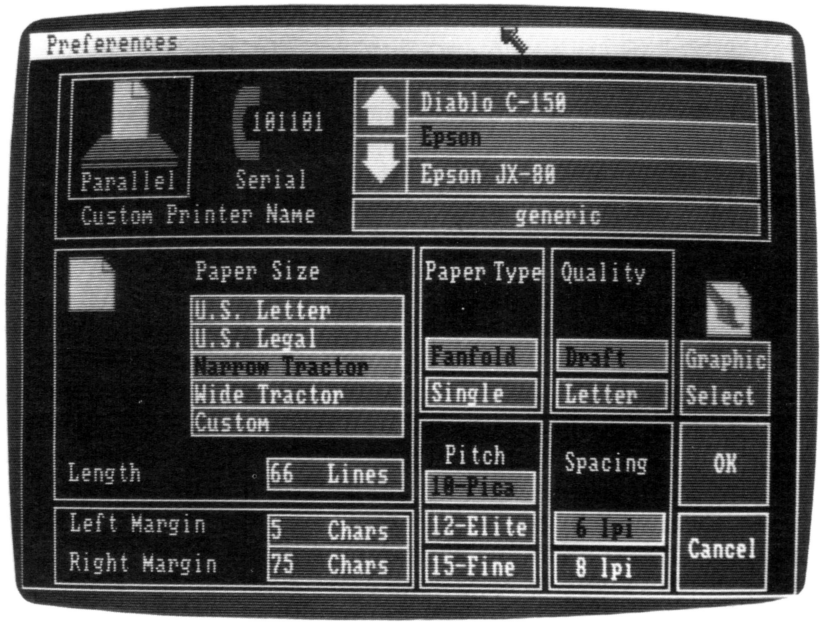
Per utilizzare il programma Preferences, è necessario digitare il comando **preferences** quando sul monitor appare l'indicazione **1>** (dopo aver inserito il dischetto di Deluxe Paint e visualizzato lo schermo CLI), invece di scrivere **dpaint** per caricare Deluxe Paint. Se state già utilizzando Deluxe Paint e desiderate ricorrere a Preferences, dovete salvare l'immagine a cui state lavorando, uscire dal programma e tornare allo schermo CLI, dove è sufficiente digitare **preferences** per caricare il programma richiesto.

Lo schermo Change Printer

Il primo schermo Preferences che appare non ha niente a che fare con il driver della stampante. Per passare ad uno schermo che consenta di modificare il driver, selezionate il riquadro denominato **Change Printer** (modifica stampante). Sul monitor apparirà lo schermo riprodotto nella Figura 3-7.

Figura 3-7.

Lo schermo Change Printer.



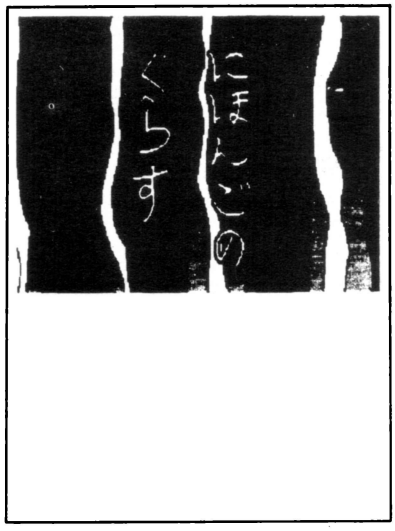
I parametri contenuti in questo schermo sono stati ideati per facilitare la stampa di testi; tuttavia alcuni influiscono anche sulla riproduzione di immagini. Le spiegazioni che seguono saranno molto utili al momento di riprodurre immagini video su carta.

Il riquadro **Margin** presente nell'angolo inferiore sinistro dello schermo consente la regolazione dei margini del testo. Misura la distanza in caratteri dal margine sinistro del foglio ai

margini sinistro e destro del testo. La spaziatura fra i caratteri è regolata dalla sezione **Pitch** situata alla destra del riquadro **Margin**. L'opzione **Pica** seleziona 10 caratteri per pollice, l'opzione **Elite** 12 e l'opzione **Fine** 15.

Le opzioni che fissano i margini possono essere utilizzate allo scopo di definire le dimensioni di un'immagine. Determinate prima la larghezza del foglio in caratteri e poi decidete dove volete che appaiano i due margini dell'immagine. Ad esempio, se usate fogli standard di 8,5" e avete selezionato l'opzione di 10 caratteri per pollice, i vostri fogli, senza margini, misureranno 85 caratteri. Se desiderate che la vostra immagine misuri 7,5" in larghezza, in modo da avere 0,5" di spazio vuoto sui due lati del foglio, dovete stabilire un margine sinistro di 5 caratteri (cioè mezzo pollice dal bordo sinistro del foglio) e un margine destro di 80 caratteri (cioè 8 pollici dal bordo sinistro e mezzo pollice dal bordo destro del foglio).

Quando il printer driver invia l'immagine alla stampante, ne conserva l'altezza entro limiti proporzionali alla larghezza stabilita. Se si stampa un'immagine utilizzando le opzioni descritte, si otterranno risultati simili all'immagine di sinistra della Figura 3-8. Qualora i margini di 5 e 80 vengano regolati a 5 e 40 caratteri, la larghezza risulta dimezzata e di conseguenza, quando si riproduce l'immagine, anche l'altezza viene dimezzata, come si vede dal confronto tra le due immagini nella Figura 3-8.



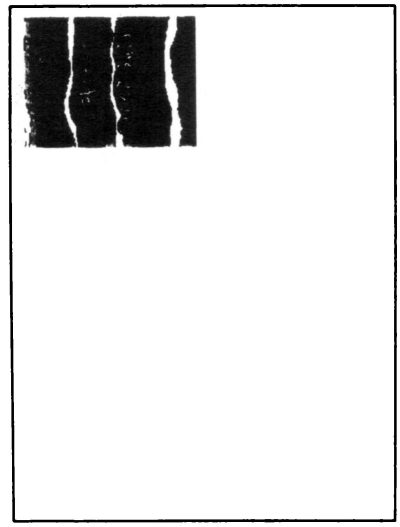


Figura 3-8.

L'immagine di sinistra è stata stampata con i margini fissati a 5 e 80. L'immagine di destra è stata stampata spostando i margini a 5 e 40, con l'effetto di rimpicciolire l'immagine a un quarto dell'originale.

Lo schermo Graphic Select

Lo schermo **Change Printer** consente di selezionare un printer driver e di regolare le dimensioni dei fogli e degli stampati. Se si desidera controllare l'aspetto e il colore della stampa è necessario selezionare il riquadro contrassegnato dalla scritta **Graphic Select** (selezione grafica). Verrà visualizzato lo schermo riprodotto nella Figura 3-9.
Il manuale *Introduzione all'Amiga* descrive le varie opzioni di questo schermo. Tuttavia possono essere utili ulteriori chiarimenti che facilitino l'utente per controllare a suo piacimento le dimensioni delle riproduzioni.

Nel riquadro **Aspect**, sulla sinistra, è possibile scegliere se stampare l'immagine orizzontalmente o verticalmente. Se si sceglie **Horizontal**, l'immagine viene riprodotta sul foglio come mostra il lato sinistro della Figura 3-10, mentre se si seleziona **Vertical** l'immagine viene stampata come si vede nell'immagine di destra.

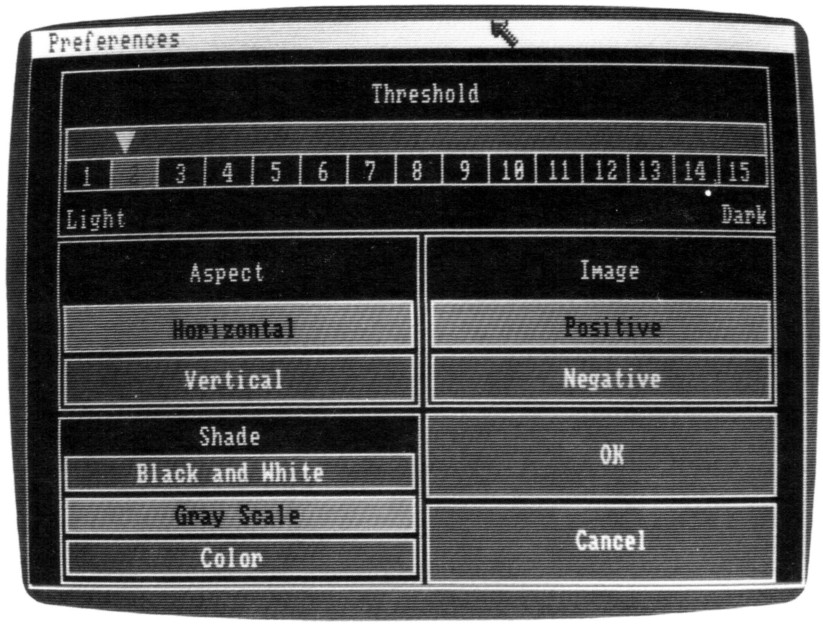


Figura 3-9.

Lo schermo Graphic Select.

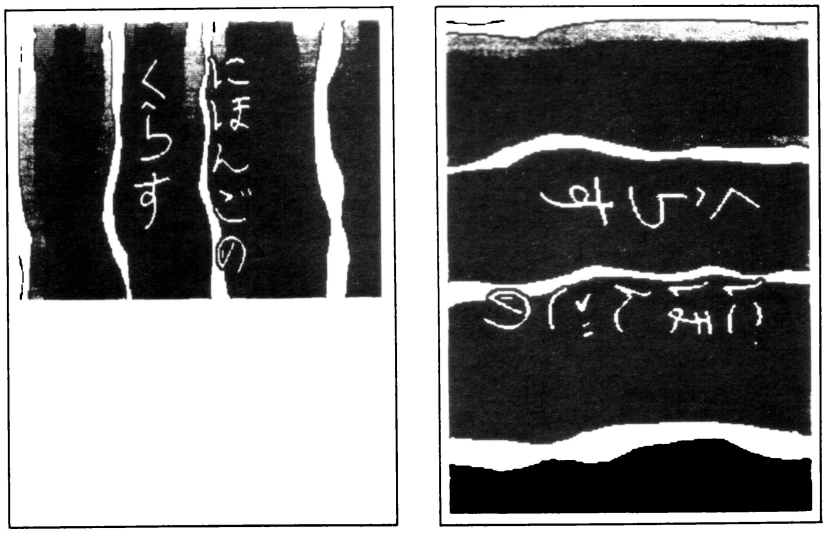


Figura 3-10.

La riproduzione di sinistra è orizzontale, quella di destra è la stessa immagine stampata verticalmente.

I due esempi illustrati nella Figura 3-10 si servono delle stesse regolazioni dei margini. Si osservi come l'opzione dei margini controlli la larghezza del lato dell'immagine che va da sinistra verso destra, qualunque esso sia. Ne consegue che una figura verticale sarà più larga della stessa figura orizzontale stampata usando la stessa regolazione dei margini (vedere Figura 3-10).

Dopo aver definito a proprio piacimento le opzioni grafiche, l'utente può selezionare **OK** per tornare al **Printer Change**, e poi selezionare ancora **OK** per immettere nuovamente sul video lo schermo **Preferences**. A questo punto, se si desidera utilizzare le opzioni appena ridefinite senza cambiare quelle preselezionate, è sufficiente selezionare **Use**, che ne consente l'uso senza doverle salvare su dischetto. Se si vogliono salvare le nuove regolazioni come dati preselezionati, scegliete **Save** in modo da utilizzarli e, in più, registrarli su disco. Quando Deluxe Paint verrà ricaricato, entreranno automaticamente in funzione le nuove regolazioni, evitando all'utente di ridefinirle attraverso il programma Preferences.

RIPRODUZIONE FOTOGRAFICA DELLE IMMAGINI

Fotografare un'immagine visualizzata sullo schermo del monitor dell'Amiga è un'operazione abbastanza semplice. Quello che serve è una macchina fotografica (preferibilmente da 35

mm. o più), un treppiede, un flessibile e una stanza buia (non una camera oscura per sviluppare le fotografie). Se si possiede un teleobiettivo poco potente (ad esempio un obiettivo da 80 mm. per una macchina da 35 mm.) l'operazione risulta facilitata, ma non è un elemento indispensabile.

Disponete la macchina fotografica sul cavalletto e posizionatela in modo che, guardando attraverso il mirino, lo schermo del monitor occupi l'intero campo visivo. Assicuratevi che la macchina sia puntata ben diritta rispetto allo schermo; se è inclinata per qualche verso la fotografia risulterà distorta. Se avete un teleobiettivo con una focale non troppo lunga, potete restare più lontani dallo schermo di quanto potete fare con un obiettivo normale, riempiendo ugualmente tutto il campo visivo del mirino. Allontanandosi dal monitor si diminuisce il grado di distorsione causato dall'incurvatura della superficie dello schermo.

Una volta posizionata e preparata la macchina fotografica, caricate l'immagine di Deluxe Paint che volete fotografare. Selezionate il pennello puntiforme dal pannello di controllo e scegliete uno dei colori utilizzati nell'immagine che volete riprodurre. Premete il tasto F8 per far scomparire la croce e muovete quindi il punto su un pixel dello stesso colore, in modo che non sia visibile. Premete il tasto F10 per far scomparire l'intestazione e il pannello di controllo. A questo punto è tutto pronto per scattare la foto.

Prima di spegnere le luci, date una ripulita allo schermo con uno straccio e un prodotto per pulire i vetri, per essere sicuri che la superficie sia pulita. Spegnete le luci. Regolate la velocità dell'otturatore a 1/15 di secondo o più e poi fissate la giusta esposizione. Utilizzate un flessibile per scattare la foto in modo da evitare di muovere la macchina; alla velocità di otturazione che state usando, la foto risulterebbe sfuocata. A questo punto non resta che far sviluppare il rullino.

ALTRI SOFTWARE GRAFICI

Deluxe Paint non è l'unico programma di grafica disponibile per l'Amiga. Esistono numerosi altri programmi che sfruttano la sua splendida grafica consentendo all'utente di creare e riprodurre immagini proprie. Per chi fosse interessato a un programma grafico, è consigliabile prendere visione, oltre che di Deluxe Paint, anche di Graphicraft e Aegis Images, in modo di scegliere quello che più si addice alla propria creatività.

GRAPHICRAFT

Graphicraft è un programma grafico messo in commercio dalla Commodore-Amiga. Con Graphicraft non è possibile creare pennelli custom multicolore o lavorare su uno schermo ad alta risoluzione, ma in ogni caso si ottengono risultati di alta qualità. È solo questione di tempo.

Un buon motivo per acquistare Graphicraft (o perlomeno dargli un'occhiata) riguarda le immagini-campione incluse nel package. Esse sfruttano appieno la capacità del programma di far ruotare i colori, in modo da simulare un battito cardiaco, una giornata di tempesta o un hot dog volante. Una volta studiati i meccanismi che consentono la creazione di tali immagini, l'operatore stesso potrà realizzare delle sequenze animate anche sofisticate, servendosi semplicemente di un programma che possa visualizzare delle rotazioni di colori.

AEGIS IMAGES

Aegis Images è un software grafico venduto insieme ad Aegis Animator, un programma di animazione video. È possibile creare immagini con Images per poi usarle nelle animazioni dell'Animator. Pur non disponendo di pennelli multicolore né di diverse risoluzioni dello schermo, Aegis Images ha altre caratteristiche speciali. Se il pezzo forte di Deluxe Paint è la vasta gamma di pennelli multicolore selezionabili per creare immagini, Images punta soprattutto sulla versatilità d'uso dei pennelli mononocromatici.

I pennelli di Images possono essere usati per dipingere motivi in tinta unita o policromi, con la possibilità di selezionare tra un'ampia gamma di funzioni per la creazione di figure geometriche come parallelogrammi e triangoli. I comandi possono essere utilizzati in più modi differenti per ottenere risultati d'effetto. Ad esempio, si può regolare un'istruzione in modo che l'ultimo punto dell'oggetto disegnato sia anche il primo dell'oggetto seguente; in tal modo gli oggetti creati risulteranno concatenati l'uno all'altro.

Images crea anche effetti molto simili a **Smear**, **Shade** e **Blend** di Deluxe Paint; "trasparency" (trasparenza) e "glow" (splendore) sono solo due esempi. Altre funzioni sono "pantograph" (pantografo) e "under" (sotto), che permettono di copiare un'immagine o di inserire un colore sotto altri già presenti sullo schermo.

LO STANDARD GRAFICO IFF

Tanto Deluxe Paint, quanto Graphicraft e Aegis Images memorizzano i file delle immagini su dischetto facendo uso dello standard grafico IFF. IFF è l'abbreviazione di Interchange File Format (Formato File Intercambiabile). Questo standard è stato sviluppato dalla Electronic Arts in collaborazione con la Commodore-Amiga, per assicurare che i file memorizzati da un

programma possano essere usati con un programma diverso. Esistono standard IFF per la grafica, la musica, il suono, i testi e tutta una gamma di altri tipi di dati che possono essere memorizzati su disco o trasmessi. Gli standard IFF lavorano con programmi diversi sullo stesso computer, ma sono progettati anche per lavorare su elaboratori differenti.

I programmatori devono far uso dello standard IFF, altrimenti i loro programmi non potranno utilizzare file provenienti da programmi diversi e memorizzati secondo tale standard. Fortunatamente, quasi tutti i programmatori di software per l'Amiga si stanno adeguando a questo standard, su specifica richiesta della Commodore-Amiga. Questo significa che l'operatore può scegliere un'immagine memorizzata con Graphicraft, Deluxe Paint o Aegis Images e caricarla in un qualsiasi programma grafico dell'Amiga che usi lo standard IFF. Ad esempio, se vi piacciono alcune caratteristiche di Deluxe Paint e per altre preferite invece Images, è possibile creare un'immagine con Deluxe Paint, memorizzarla e caricarla in Images per gli ultimi ritocchi.

HARDWARE GRAFICO PER L'AMIGA

Se vi interessano particolarmente le applicazioni grafiche, allora vale la pena di usare un hardware che permetta di esprimere al massimo le capacità grafiche dell'Amiga. I prossimi paragrafi contengono un elenco di vari prodotti hardware grafici per Amiga, unitamente ad alcuni consigli per la scelta delle attrezzature adeguate alle vostre necessità.

MONITOR

Il monitor è senza dubbio l'elemento hardware più usato. Se volete elevare la qualità delle immagini e possedete un monitor monocromatico o composito, potete acquistare un monitor RGB; ma prima di comprarlo è bene tenere presente tutta una serie di fattori.

Esistono innanzitutto due tipi di monitor RGB: analogici e digitali. I monitor RGB digitali sono quelli più usati con i PC IBM e quindi i più facilmente reperibili. L'Amiga può visualizzare immagini su un monitor RGB digitale, ma questo tipo di monitor può visualizzare solo 16 colori, limitando così lo spettro cromatico dell'Amiga. Un monitor RGB analogico può invece visualizzare una gamma di colori completa ed è decisamente il miglior monitor RGB per la grafica a colori.

Sono disponibili diversi tipi di monitor RGB analogici. Al momento della scelta è consigliabile assicurarsi che il monitor disponga di una risoluzione verticale di almeno 512 linee, per visualizzare opportunamente le immagini ad alta risoluzione dell'Amiga. Dovreste anche controllare la risoluzione dei punti sul monitor; più piccoli sono i punti, più nitida sarà l'immagine sullo schermo. Un comune monitor RGB può avere una risoluzione dei punti di circa 0,4 mm. Un monitor particolarmente nitido può arrivare a 0,28 mm.

Se non siete disposti a spendere troppo per un monitor dedicato al computer, potete acquistare un apparecchio televisivo in grado di visualizzare i segnali RGB analogici. In questo modo sarete in possesso di un monitor e di un vero e proprio televisore. Il Sony KV1311 è un esempio di TV/monitor di questo tipo.

Un'altra alternativa consiste nell'acquistare uno dei tanti monitor video in commercio in grado di lavorare con un sistema videocomposito che abbia anche un'entrata per un segnale RGB analogico. Di solito questi monitor non dispongono di un sintonizzatore, ma quasi tutti riescono a ricevere un segnale da un videoregistratore; potete quindi usare il sintonizzatore di quest'ultimo per ricevere i programmi televisivi sul monitor. Alcuni monitor hanno uno schermo di 25 o 26 pollici, ottimo per la presentazione delle immagini dell'Amiga a un folto pubblico di spettatori e per giocare con videogame interminabili standosene comodamente in poltrona.

Chi desiderassse risolvere il problema della vibrazione del display in alta risoluzione (640x512 pixel), deve invece orientarsi su un monitor ad alta persistenza.

STAMPANTI

Una stampante a colori è fondamentale per chi desidera riprodurre su carta le immagini. Il software sistema dell'Amiga dispone di tre driver per tre diverse stampanti a colori: la Okimate 20, la Epson JX-80 e la Diablo C-150. Per usare altre stampanti a colori l'Amiga deve essere prima munito di un apposito driver.

La stampante Okimate 20

La Okimate 20, prodotta dalla Okidata, è una stampante a colori a basso prezzo. È una stampante termica: utilizza un nastro coperto di cera e una testina termica che riscalda il colore trasferendolo dal nastro alla carta. Come risultato di questa tecnica, i colori risultano molto saturi.

La Okimate 20 presenta due grossi svantaggi. Consuma molto velocemente il nastro, che deve essere sostituito dopo essere stato usato una sola volta. La registrazione della testina non è sempre precisa; le immagini presentano spesso strisce

bianche che dividono i tratti colorati che formano l'immagine. Ciò nonostante va tenuto presente che, per quel poco che costa, anche questa stampante ha il suo valore.

La stampante Epson JX-80

La Epson JX-80 è una stampante a impatto con un nastro multicolore per imprimere la carta. La JX-80 è probabilmente la migliore delle tre stampanti a colori che stiamo considerando, in quanto può anche riprodurre testi in letter quality, se munita di un apposito chip. Accetta qualsiasi tipo di carta, o quasi, e continua a usare lo stesso nastro finché non ha perso tutto l'inchiostro.

Purtroppo con la JX-80, come con quasi tutte le stampanti a impatto a colori, si ottengono riproduzioni su carta standard che appaiono sbiadite rispetto alle immagini sullo schermo. D'altra parte, la JX-80 può essere usata tanto per l'elaborazione dei testi quanto per la stampa grafica, il tutto a un prezzo ragionevole.

Le stampanti Diablo C-150 e Xerox 4020

Se siete seriamente interessati alla riproduzione su carta delle vostre opere grafiche a colori, allora dovreste scegliere fra una stampante Diablo a getto d'inchiostro, prodotta dalla Xerox, e una stampante Xerox 4020 a getto d'inchiostro, versione di qualità superiore che usa lo stesso driver della C-150. Deve essere una scelta oculata in quanto si tratta di stampanti costose in grado di offrire colori particolarmente vivi e un'estrema cura dei singoli particolari.

Le stampanti a getto d'inchiostro spruzzano inchiostro colorato su speciali fogli di carta patinata che non assorbe né lascia scivolare il colore. I punti d'inchiostro sono così piccoli che si possono ottenere risoluzioni eccellenti. Gli inchiostri sono relativamente densi, in modo che non penetrino nel foglio; i colori risultano così ben saturi. È inoltre possibile utilizzare fogli di carta da lucido o acetato per ottenere immagini colorate su sfondo trasparente; in questo modo un programma grafico per Amiga può essere utilizzato per creare e stampare grafici commerciali, presentati poi con una lavagna luminosa.

Queste due stampanti non sono invece molto pratiche per la riproduzione di testi in quanto, pur producendo caratteri di buona qualità, possono stamparli solo sui fogli di carta speciale patinata, non particolarmente indicati per lettere commerciali o corrispondenza. Affinché la C-150 produca sempre immagini con colori vivi, è necessario trattarla con molta cura, sciacquando la bocchetta dello spruzzatore e pulendo di tanto in tanto le bolle d'inchiostro che si formano. La 4020, invece, non crea problemi di manutenzione e produce colori più saturi di quelli

della C-150. A parte la manutenzione e il costo, queste stampanti sono senza dubbio le migliori presenti sul mercato e vale la pena acquistarle se avete intenzione di usarle anche a livello professionale.

IL DIGITALIZZATORE DI IMMAGINI AMIGA LIVE!

Con il framegrabber (letteralmente "acchiappa inquadrature") Amiga LIVE!, l'utente può aggiungere alla sua collezione immagini tratte dalla vita di tutti i giorni, trasformando le immagini di una telecamera, di un televisore, di un videoregistratore, di un altro computer o di qualsiasi altra fonte video in un'immagine Amiga a bassa risoluzione.

L'Amiga LIVE! è predisposto per essere inserito nella porta di espansione situata sul lato destro dell'Amiga 1000. Su un lato del box dell'Amiga LIVE! è presente un jack nel quale è possibile inserire un cavo video standard per trasferire un'immagine da un'altra fonte video. Quando accendete l'Amiga, mettete in azione un dischetto che contiene il driver per l'Amiga LIVE!, cioè il software in grado di leggere le immagini del digitalizzatore. Caricato il driver, è possibile vedere sullo schermo le immagini prodotte dal framegrabber.

Amiga LIVE! è un digitalizzatore in tempo reale. In altre parole riesce a ricevere un'immagine video e a trasformarla in un'immagine Amiga a bassa risoluzione aggiornando la visualizzazione 12 volte al secondo. Se la fonte è un'immagine animata, il risultato sarà una serie di immagini in movimento sullo schermo. È possibile fermare l'immagine in qualsiasi momento. L'Amiga LIVE! riesce a catturare immagini da un video a colori utilizzando sul proprio schermo 32 colori, oppure riproducendo l'immagine in bianco e nero usando altre 16 tonalità di grigio. Le immagini possono poi essere memorizzate su dischetto; il software le registra con lo standard IFF, dando così la possibilità all'utente di poter in seguito caricare le immagini in un programma grafico che utilizza lo stesso standard e modificarle a piacimento.

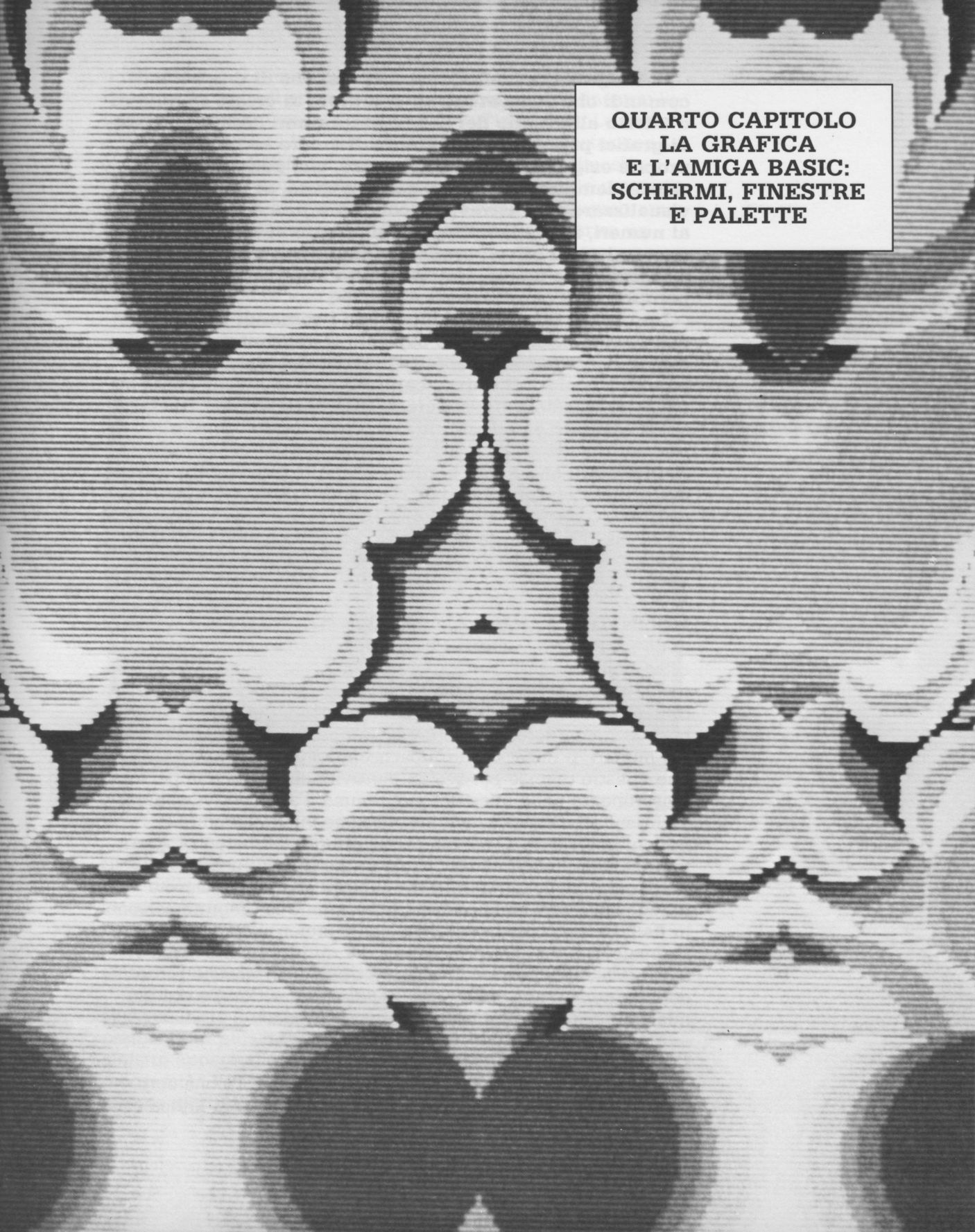
Amiga LIVE! non è un semplice digitalizzatore di immagini ferme. Dal momento che opera in tempo reale e che l'Amiga invia istantaneamente le sue immagini, le immagini digitalizzate dall'Amiga LIVE! possono essere immesse in un videoregistratore, allo scopo di creare originali creazioni video ricche di stupendi effetti speciali. L'Amiga LIVE! svolge la funzione di elaboratore di immagini, consentendo di "congelare" un singolo fotogramma, modificare i colori e creare molti altri effetti speciali tipici delle produzioni basate sull'uso del video, come ad esempio i videoclip musicali.

IL VIDEODIGITALIZZATORE DIGIVIEW

Digiview è un pacchetto hardware/software che fa uso di una telecamera per trasferire immagini a colori e monocromatiche nella memoria dell'Amiga. L'hardware si inserisce direttamente nella porta parallela del computer e contiene un singolo jack video RCA. L'unico altro "add-on" hardware fornito è un filtro colore circolare, completo di un apposito supporto che consente la sua installazione davanti all'obiettivo della videocamera, il quale permette di separare i colori rosso, verde e blu dell'immagine video. Dal momento che non è possibile utilizzare il filtro unitamente a un videoregistratore, non si possono ottenere immagini a colori da questo tipo di fonte video.

Il software fornito con Digiview è di ottima qualità e si distingue soprattutto per la completezza dei comandi disponibili, attraverso i quali si è in grado di controllare ogni aspetto dell'immagine nelle varie fasi della sua formazione. È infatti possibile regolare la luminosità, il contrasto, la saturazione, l'intensità di ogni componente cromatica e la messa a fuoco dell'immagine stessa, che può essere registrata come file in formato IFF in 32 colori o come immagine Hold & Modify in 4096 colori. Si può utilizzare sia la bassa che l'alta risoluzione e le immagini b/n sono definite da 128 sfumature di grigio.

Nelle pagine precedenti vi è stato fornito un quadro delle funzioni di Deluxe Paint e delle modalità per la riproduzione su carta o pellicola delle vostre creazioni. Vi sono stati inoltre presentati alcuni degli altri programmi grafici disponibili per l'Amiga. I programmatori in BASIC, o chiunque si interessi degli aspetti più prettamente tecnici della grafica dell'Amiga, non possono fare a meno, a questo punto, di passare alla lettura dei successivi tre capitoli, che trattano i comandi grafici dell'Amiga BASIC.



QUARTO CAPITOLO
LA GRAFICA E L'AMIGA BASIC: SCHERMI, FINESTRE E PALETTE

L'Amiga BASIC della Microsoft dispone di una serie di comandi che consentono all'operatore di creare immagini d'effetto all'interno dei propri programmi BASIC. I comandi grafici permettono all'utente di creare secondo le proprie esigenze schermi e finestre, che possono poi essere riempite con figure, colori e testo. Si possono visualizzare dati attraverso colori e figure senza ricorrere ai numeri, richiedere l'input dell'utente avvalendosi di una varietà di effetti grafici e aggiungere immagini caratterizzate da una grafica molto accurata anche solo per motivi di carattere estetico.

I prossimi tre capitoli illustrano i comandi grafici dell'Amiga BASIC. Questo capitolo si concentra sui comandi necessari all'utente per creare finestre, schermi e palette di colori. Il quinto capitolo spiega invece come si possono tracciare linee e creare figure dai colori diversi. Il sesto capitolo è dedicato ai concetti fondamentali che permettono di aggiungere testi alle immagini, di adeguare la grafica alle diverse misure delle finestre e di tagliare diverse sezioni grafiche per riassemblarle in un nuovo ordine.

INTRODUZIONE ALL'AMIGA BASIC

Se non avete mai utilizzato l'Amiga BASIC, impiegherete un po' di tempo per abituarvi alle sue particolari caratteristiche. A differenza della maggior parte delle versioni del BASIC utilizzate su altri computer, il BASIC dell'Amiga non richiede l'impiego dei numeri di linea. Un'altra particolarità consiste nell'uso di due finestre diverse per svolgere attività separate: una per l'inserimento delle linee del programma, l'altra per visualizzarne i risultati. Questa breve introduzione vuole illustrare le nozioni necessarie a immettere, far girare e fermare i programmi in Amiga BASIC, allo scopo di permettervi di usare i programmi modello riportati nei capitoli successivi. Questa introduzione non ha la pretesa di insegnare la programmazione in BASIC – un compito specifico che è preferibile affidare a un manuale di BASIC per principianti o a un insegnante – ma vi sarà certamente di grande utilità pratica. Informazioni più dettagliate sull'uso dell'Amiga BASIC vengono fornite dal manuale allegato alla confezione del programma.

LE FINESTRE LIST E OUTPUT

Per avviare l'Amiga BASIC è sufficiente aprire l'icona corrispondente proprio come in qualsiasi altro caso: la si indica con

il puntatore e si preme due volte il tasto Select sul mouse. Dopo pochi secondi appariranno sullo schermo due finestre: la finestra List e la finestra Output (vedere Figura 4-1). La prima è intitolata appunto **LIST**, mentre la seconda ha l'intestazione (almeno per ora) **BASIC**. La linea d'intestazione della finestra Output riporta il nome del programma che si trova in memoria in quel momento. Non avendo ancora caricato alcun programma, il titolo è semplicemente **BASIC**. Non appena si carica un programma da un dischetto o si salva un nuovo programma sotto un nome qualsiasi, l'intestazione della finestra Output riporterà il nome di tale programma.

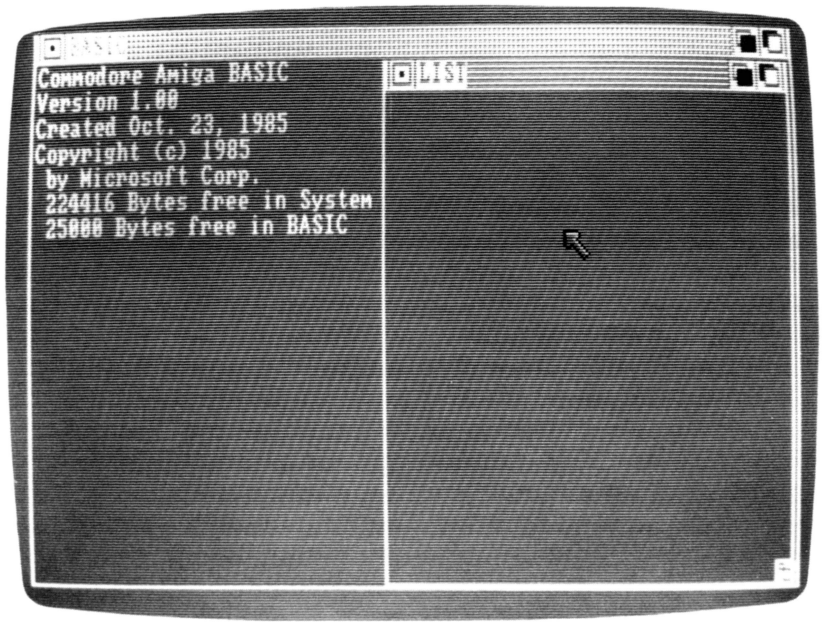


Figura 4-1.

Le finestre List e Output dell'Amiga BASIC.

Le due finestre tengono separate le linee in BASIC dai risultati del programma. Le linee vanno immesse e immagazzinate nella finestra List. Quando si fa girare il programma, tutti i risultati, siano testi o immagini, appaiono generalmente nella finestra Output.

IMMISSIONE DI UN PROGRAMMA NELLA FINESTRA LIST

Per immettere un programma nella finestra List è necessario selezionare la finestra muovendo il puntatore al suo interno e premendo il pulsante Select del mouse. Nell'angolo superiore sinistro appare allora un cursore, una lineetta verticale. Il programma va immesso battendolo linea per linea e premendo il tasto RETURN al termine di ogni linea. Dal momento che le

righe non vengono numerate, è importante mantenerle nel giusto ordine.

Se si arriva all'estremità destra della finestra List prima di aver finito di digitare una linea molto lunga, il contenuto della finestra scorrerà automaticamente verso sinistra per permettere all'utente di proseguire l'immissione dei dati. Non appena si preme RETURN, il contenuto tornerà nella posizione originale, all'estremità sinistra della finestra List. Analogamente, si verificherà uno scroll di ritorno verso l'alto qualora sia stato raggiunto e superato il margine inferiore della finestra.

È possibile correggere gli errori e apportare modifiche ai programmi, semplicemente riportando il cursore sulle linee già immesse. I quattro tasti indicati dalle freccette muovono il cursore nelle quattro direzioni senza alterare i caratteri che toccano nei loro spostamenti. È inoltre possibile utilizzare il puntatore del mouse per muovere il cursore, indicando semplicemente il punto in cui si vuole appaia il cursore e premendo il tasto Select. Raggiunte le istruzioni che si desiderano modificare, si inizia a battere il testo che si vuole inserire o si usa il tasto BACKSPACE per cancellare i dati errati che precedono il cursore. Le linee possono anche essere cancellate evidenziandole prima col mouse e premendo poi il tasto BACKSPACE.

LO SCROLL DEI LISTATI

Se il numero delle linee di un programma supera la quantità contenibile nella finestra List, è possibile far scorrere il listato pagina per pagina, tenendo premuto uno dei tasti SHIFT e muovendosi con i due cursori di spostamento verticale; altrimenti è possibile saltare direttamente all'inizio o alla fine del programma, tenendo abbassato uno dei tasti ALT e premendo contemporaneamente uno dei tasti che muovono il cursore verticalmente. Se non si riesce a visualizzare le linee in tutta la loro lunghezza, è possibile allargare la finestra List spostandola verso sinistra e utilizzare poi il gadget di dimensionamento posto nell'angolo inferiore destro per prolungare la finestra verso destra. Se anche questa operazione dovesse rivelarsi inutile, usate il tasto SHIFT insieme ai cursori di spostamento laterale per far scorrere totalmente la finestra verso destra o verso sinistra, oppure il tasto ALT con i cursori destro e sinistro per spostarsi all'inizio o alla fine di una linea.

COME FAR GIRARE UN PROGRAMMA

Dopo aver immesso un programma in memoria, lo si può far girare selezionando **Start** dal menu **Run**. La finestra List sparisce e appaiono i risultati del programma nella finestra Output, siano essi testi o immagini. Terminato il programma, riappare la finestra List. Se il programma non si ferma da solo, è possibile interromperlo selezionando **Stop** dal menu **Run**, oppure premendo

contemporaneamente il tasto Amiga di destra e il tasto del punto (.), o ancora tenendo abbassato CTRL e premendo C. Se quando il programma termina la finestra List non riappare immediatamente, selezionate **Show List** dal menu **Windows** oppure, più semplicemente, premete il tasto Amiga di destra insieme al tasto L.

IMMISSIONE DI COMANDI DIRETTI NELLA FINESTRA OUTPUT

Se selezionate la finestra Output con il mouse, è possibile utilizzarla anche per l'immissione di comandi BASIC. Anziché immagazzinare i comandi, come succede nella finestra List, l'Amiga BASIC esegue ogni comando immesso nella finestra Output non appena l'utente preme il tasto RETURN. In questo modo, ogni comando diventa un comando diretto, in quanto l'utente può verificarne immediatamente i risultati. La finestra Output è utile, ad esempio, per controllare l'effetto di un comando prima di inserirlo in un programma.

MEMORIZZAZIONE DI UN PROGRAMMA

Per memorizzare un programma è sufficiente selezionare **Save** dal menu **Project**. L'Amiga BASIC richiederà di immettere un nome con cui salvare il programma. A questo punto puntate il riquadro col mouse e premete il pulsante per ottenere un cursore, dopodiché immettete il nome desiderato e premete RETURN. Il programma sarà così memorizzato su un dischetto sotto forma di file BASIC. Il nome del programma apparirà nell'apposito spazio della finestra Output. Mentre modificate il programma potete continuare a usare **Save**; così facendo l'Amiga BASIC salverà il programma modificato con lo stesso nome usato prima, cancellando la versione precedente. Per memorizzare un programma sotto un nome differente, dovete selezionare **Save As** dal menu **Project** e immettere il nuovo nome. Il nome visualizzato dalla finestra Output verrà sostituito da quello nuovo, ma il vecchio file resterà sul dischetto senza essere cancellato.

Per caricare un programma da disco bisogna selezionare **Open** dal menu **Project**, puntare il box e premere il tasto del mouse, immettere il nome del programma desiderato e premere RETURN. Se il programma a cui state lavorando non è stato memorizzato con le ultime modifiche, il computer vi avvertirà, consentendovi di memorizzare o meno il nuovo lavoro. Dopodiché l'Amiga BASIC immetterà il programma da voi richiesto cancellando qualsiasi cosa si trovi nella finestra List. La stessa cosa succede quando l'utente immette **New** per iniziare a scrivere un nuovo programma.

COME USCIRE DAL BASIC

Quando si vuole uscire dal BASIC per tornare al Workbench è sufficiente selezionare **Quit** dal menu **Project** o chiudere entrambe le finestre List e Output selezionando i rispettivi gadget di chiusura. Se invece l'utente vuole semplicemente entrare nel Workbench senza uscire dal BASIC (ad esempio, per verificare se un dischetto è già pieno), potrà selezionare i gadget di ritorno (back gadget) delle due finestre del BASIC, facendole sparire dietro le altre finestre del Workbench. Controllato quanto desiderava, l'utente potrà riportare sullo schermo le finestre List e Output selezionando i loro gadget di avanzamento (front gadget) oppure i back gadget delle finestre del Workbench. Talvolta, per fare riapparire la finestra List, è necessario selezionare **Show List** dal menu **Windows**.

INTRODUZIONE A SCHERMI, FINESTRE E PALETTE

Abbiamo già visto nel secondo capitolo che l'Amiga crea le immagini servendosi di pixel, minuscoli "elementi costitutivi dell'immagine". Osservando da vicino qualsiasi immagine o testo visualizzato sul monitor dell'Amiga, è possibile distinguere i pixel chiaramente: sono quei minuscoli quadratini che formano le immagini. Per costruire un oggetto sullo schermo del monitor, l'Amiga ordina una serie di singoli pixel colorandoli uno per uno in modo da dare forma all'oggetto. Visti anche da pochi centimetri, i pixel si combinano tanto perfettamente da sembrare un unico oggetto compatto.

La dimensione dei pixel utilizzati dall'Amiga per creare un'immagine determina il tipo di risoluzione. In un'immagine ad alta risoluzione i pixel sono piccoli e non facilmente distinguibili, permettendo così di riprodurre linee curve e diagonali in modo abbastanza scorrevole e realistico. In un'immagine a bassa risoluzione i pixel sono invece più grandi e passano meno inosservati; diagonali e curve risultano frastagliate ed è più facile distinguerne i singoli punti.

SCHERMI

L'Amiga offre quattro diversi tipi di risoluzione per la visualizzazione di immagini e testi. La risoluzione può essere scelta al momento della creazione di uno schermo (procedura illustrata nella sezione "Il comando screen" di questo stesso capitolo). È possibile avere un massimo di cinque

schermi sullo stesso monitor, ognuno con una risoluzione diversa. Ogni schermata dell'Amiga si estende per tutta la larghezza del quadro e ha una propria linea d'intestazione. È possibile creare uno schermo che non sia visualizzato sull'intera ampiezza del quadro, ma in ogni caso esso controllerà tutta la superficie. La linea d'intestazione può essere afferrata con il puntatore del mouse, per far scorrere lo schermo verso l'alto o verso il basso (l'utente deve tenere premuto il tasto Select e muovere contemporaneamente il cursore nella direzione desiderata). Questa operazione fa apparire lo schermo immediatamente dietro a quello che state usando (se ce n'è uno); anche questo nuovo schermo può essere fatto scorrere, così come se ne può afferrare l'intestazione per muoverlo verso l'alto o verso il basso.

La linea d'intestazione dello schermo è usata anche per visualizzare i menu. Tenendo premuto il tasto Menu del mouse, viene riportato nell'intestazione l'elenco dei vari menu. Per farne apparire uno è sufficiente indicarlo col mouse mentre il tasto Menu è ancora abbassato.

La linea d'intestazione comprende anche i front e back gadget, posti nell'angolo di destra. Se sono presenti sul monitor più schermi contemporaneamente, risultano stratificati, cioè uno copre l'altro. Selezionare il gadget di sinistra (back gadget) significa spostare lo schermo visibile in quel momento dietro a tutti gli altri; selezionare il gadget di destra (front gadget) significa invece porre quello schermo sopra a tutti gli altri.

Un buon esempio di schermo è quello del Workbench. Quando avviate l'Amiga, lo schermo del Workbench appare con l'intestazione **Workbench release 1.2** (o versioni successive) e con le icone dei dischi presenti nei drive. È possibile far scorrere lo schermo verso l'alto o verso il basso puntando la linea d'intestazione con l'indicatore del mouse, tenendo premuto il tasto Select e muovendo il mouse nella direzione desiderata. Se provate a usare i gadget, non succede assolutamente nulla, perché non ci sono altri schermi sul monitor. Premendo il tasto Menu del mouse, appariranno i menu del Workbench. Lo schermo del Workbench con i suoi gadget è riprodotto nella Figura 4-2 (riportata a pagina 103).

FINESTRE

Gli schermi forniscono superfici di diverse risoluzioni. In ogni singolo schermo le finestre permettono di creare aree di attività ben definite. Se l'utente intende realizzare grafica o visualizzare testi sullo schermo, può farlo solo per mezzo di una finestra: non si possono visualizzare immagini o testi su uno schermo privo di finestre. Le finestre devono avere la stessa risoluzione e gli stessi colori dello schermo a cui appartengono, pertanto non è possibile creare nello stesso schermo finestre con diverse risoluzioni o colori.

A differenza degli schermi, le finestre non devono necessariamente coprire l'intera estensione del monitor. Possono essere di qualsiasi dimensione, purché possano essere contenute dallo schermo a cui appartengono. Ogni finestra ha limiti precisi e può avere una linea d'intestazione. La linea d'intestazione può visualizzare il nome della finestra, nonché includere tre gadget opzionali: i consueti back e front gadget, più un gadget di chiusura. Selezionare quest'ultimo significa far scomparire la finestra dallo schermo. Selezionare il back gadget equivale a muovere la finestra dietro a tutte le altre finestre di quello schermo. Selezionando infine il front gadget si ottiene lo spostamento della finestra davanti a tutte le altre.

Nell'angolo inferiore sinistro della finestra è presente un gadget opzionale per il dimensionamento. Indicando col mouse questo gadget mentre si tiene premuto il tasto Select, e poi muovendo il mouse, è possibile modificare le dimensioni della finestra. Se la finestra ha una linea d'intestazione ed è più piccola dello schermo che la contiene, è possibile spostare l'intera finestra all'interno dei limiti dello schermo, semplicemente puntando la linea d'intestazione, tenendo premuto il tasto Select e muovendo il mouse.

Aprendo l'icona del Workbench che appare sullo schermo, come nella Figura 4-2, si può avere un ottimo esempio di finestra con tanto di gadget, intestazione e piena mobilità. La finestra può essere spostata in qualsiasi punto dello schermo, se ne possono ridefinire le dimensioni e, se si vuole, la si può chiudere. Quando lo schermo del Workbench viene spostato, anche le finestre si spostano, seguendone i movimenti.

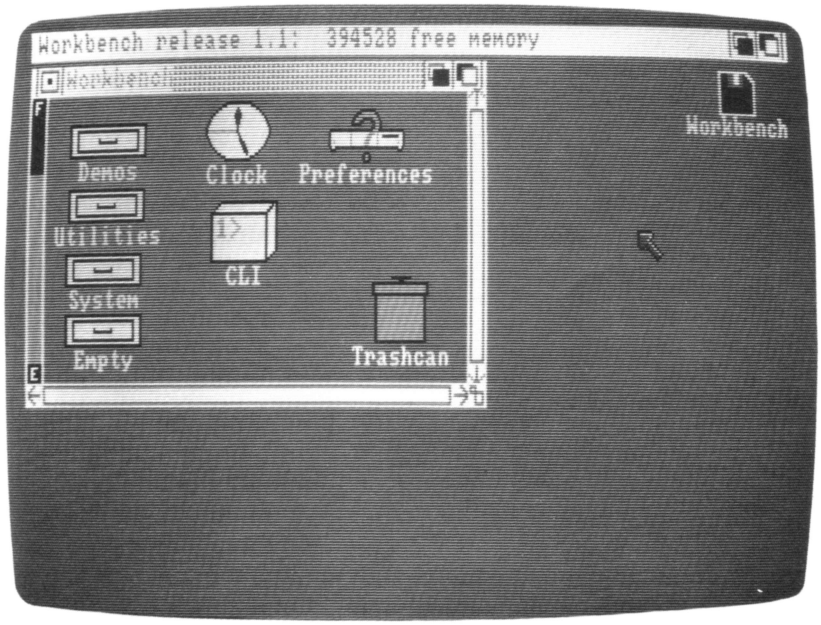


Figura 4-2.

Una finestra dello schermo del Workbench che mostra il contenuto del disco.

PALETTE

Ogni schermo dell'Amiga ha una propria gamma di colori detta palette (tavolozza). Una palette può avere un massimo di 32 colori, selezionabili da uno spettro cromatico di ben 4096 tonalità differenti, sufficienti a creare una palette esclusiva per ciascuno schermo. Le procedure che consentono di modificare il contenuto di una palette dal BASIC sono spiegate in una sezione successiva di questo capitolo.

L'Amiga fa uso delle tinte della palette per colorare i pixel dello schermo o della finestra sullo schermo che utilizzi quei colori. Ad esempio, il primo colore della palette è usato per colorare lo sfondo dello schermo e delle sue finestre, il secondo colore per i margini delle finestre e le linee d'intestazione. Gli altri due colori della palette sono usati per evidenziare in campo inverso le opzioni selezionate dai menu.

Quando l'operatore avvia il Workbench, viene visualizzato uno schermo con fondo blu e margini e intestazioni delle finestre bianchi, sempre che i colori non siano già stati cambiati. Quando l'utente seleziona un'opzione di un menu, l'opzione interessata cambia colore, da blu a nero e da bianco ad arancione. Se l'operatore ricorre al programma Preferences per modificare i colori del Workbench, modifica l'intera palette dello schermo, ottenendo una nuova colorazione per lo sfondo, i margini e le opzioni selezionate in campo inverso.

IL COMANDO SCREEN

Iniziando a lavorare con l'Amiga BASIC, si utilizzano le finestre List e Output che il BASIC crea sullo schermo del Workbench. Qualsiasi figura o testo creato appare sulla finestra Output, con la risoluzione dello schermo del Workbench e un massimo di quattro colori. Volendo utilizzare più colori e una diversa risoluzione, è possibile creare un nuovo schermo con il comando SCREEN. La sintassi di SCREEN è:

SCREEN *numero di ID dello schermo, larghezza, altezza, profondita', risoluzione*

I cinque parametri che seguono il comando SCREEN devono apparire sempre in questo ordine e devono essere separati da virgole.

IL NUMERO DI ID(entificazione) DELLO SCHERMO

Oltre allo schermo del Workbench, è possibile creare altri quattro schermi con l'Amiga BASIC e avere così sul monitor contemporaneamente un totale di cinque schermi. A ognuno di essi viene assegnato un numero di ID, cioè di identificazione: 1, 2, 3 o 4. Lo schermo del Workbench è sempre indicato con -1. Come verrà spiegato in seguito, quando si utilizzerà il comando WINDOW per creare una finestra, il numero di ID indicherà al BASIC lo schermo in cui la finestra deve essere aperta.

IL MODO DI RISOLUZIONE DELLO SCHERMO

Anche se il modo di risoluzione è l'ultimo parametro del comando SCREEN, è necessario decidere quale risoluzione si intende usare prima di stabilire la larghezza, l'altezza e la profondità dello schermo. Si può scegliere uno qualsiasi dei quattro modi di risoluzione disponibili. Ogni modo utilizza una diversa dimensione di pixel.

Vi sono due diverse larghezze di pixel: grossa, a bassa risoluzione, e sottile, ad alta risoluzione. In uno schermo ad alta risoluzione si possono inserire 640 pixel in senso orizzontale. I pixel a bassa risoluzione hanno una larghezza doppia rispetto a quelli ad alta risoluzione; uno schermo a bassa risoluzione contiene quindi in tutta la sua larghezza 320 pixel.

Ci sono anche due diverse altezze di pixel: alti, o non-interlace, e bassi, o interlace. Si possono inserire nello schermo 200 (256 negli Amiga europei) pixel non-interlace in senso verticale. I pixel interlace sono alti la metà di quelli non-interlace e in uno schermo se ne possono inserire 400 (512 nei modelli europei).

Combinando le due larghezze e le due altezze dei pixel, si ottengono quattro diverse risoluzioni, che possono essere specificate con i seguenti numeri di modo di risoluzione:

- **Modo 1**: bassa risoluzione, pixel non-interlace con un massimo di 320 per 200 (256) pixel su uno schermo.
- **Modo 2**: alta risoluzione, pixel non-interlace con un massimo di 640 per 200 (256) pixel su uno schermo.
- **Modo 3**: bassa risoluzione, pixel interlace con un massimo di 320 per 400 (512) pixel su uno schermo.
- **Modo 4**: alta risoluzione, pixel interlace con un massimo di 640 per 400 (512) pixel su uno schermo.

Lo schermo del Workbench corrisponde al modo 2.

Gli schermi che utilizzano i modi 3 e 4 tremano leggermente sul monitor perché usano pixel interlace. Per inserire 400 (o 512) linee di pixel sullo schermo, l'Amiga utilizza un metodo di visualizzazione chiamato indirizzamento interlace, che effettua prima la scansione di tutte le linee dispari del quadro e poi di tutte le linee pari. Il passaggio fra le linee pari e quelle dispari avviene circa ogni 1/30 di secondo, velocità sufficientemente lenta a provocare un leggero tremolio. I pixel non-interlace, invece, non tremano, perché l'Amiga traccia tutte le linee in un solo passaggio che dura 1/60 di secondo, velocità che consente di evitare tremolii.

Se ci sono più schermi su un unico monitor e ognuno di essi utilizza l'indirizzamento interlace, tutta l'immagine sul monitor tremerà leggermente, perché tutto il quadro è occupato da pixel interlace. L'indirizzamento interlace non modifica le dimensioni dei pixel degli schermi non-interlace. A parte il tremolio, essi appaiono esattamente come sono. Chiudendo tutti gli schermi interlace si eliminerà la vibrazione sul monitor. Il tremolio dello schermo può essere evitato utilizzando un monitor ad alta persistenza.

LA LARGHEZZA E L'ALTEZZA DELLO SCHERMO

Dopo aver deciso il modo di risoluzione, è possibile stabilire le dimensioni dello schermo fissando la larghezza e l'altezza in pixel, tenendo conto dei limiti posti dalla risoluzione prescelte. Ad esempio, non è possibile creare uno schermo in modo 1 con 400 per 200 pixel, poiché esso ha un'ampiezza massima di 320 per 200 (256).

Se si crea uno schermo più piccolo rispetto alle massime dimensioni consentite, l'Amiga BASIC lo colloca nell'angolo inferiore sinistro del monitor. Ad esempio, uno schermo in modo 1 con una larghezza ed un'altezza di 160 per 100 apparirà

sul monitor come nella Figura 4-3. È possibile spostare questo schermo più in alto o più in basso rispetto alla sua posizione originale. Noterete che la parte sulla destra dello schermo, pur essendo vuota, si sposterà comunque in alto o in basso seguendo i movimenti dello schermo, dal momento che la risoluzione prescelta copre tutta la superficie del monitor.

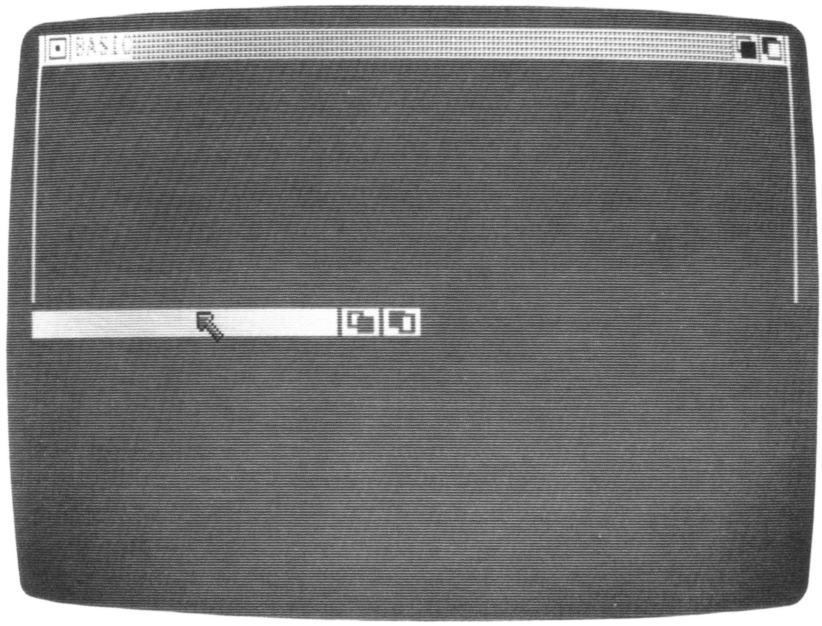


Figura 4-3.

Schermo in modo 1 di 160 per 100 pixel.

LA PROFONDITÀ DELLO SCHERMO

La profondità dello schermo viene misurata in bit-plane. Il numero di bit-plane determina il numero di colori che si possono utilizzare sullo schermo e nelle finestre. Ogni bit-plane è una parte di RAM dell'Amiga che utilizza un bit per memorizzare le informazioni riguardanti il colore di ogni pixel dello schermo. Unendo i bit-plane, l'Amiga può utilizzare più bit per memorizzare il colore di ogni pixel.

Creando uno schermo si possono specificare in qualunque momento da uno a cinque bit-plane. Uno schermo con un bit-plane ha solo due colori, poiché c'è un solo bit in memoria per ogni pixel (il bit può essere acceso o spento, quindi due colori). Ogni bit-plane che viene aggiunto allo schermo inserisce un altro bit in memoria per ogni pixel, offrendo così una gamma più vasta di colori per la visualizzazione. Ad esempio, uno schermo a 2 bit-plane assegna due bit a ogni pixel, offrendo una scelta di quattro colori, dato che due bit permettono quattro combinazioni di zero e uno. La tabella

seguente mostra quanti colori possono essere utilizzati dopo aver stabilito la profondità dello schermo:

Numero di bit-plane	Numero di colori
1	2
2	4
3	8
4	16
5	32

Nei modi di risoluzione 1 e 3 è possibile specificare fino a cinque bit-plane, mentre nei modi 2 e 4 c'è un limite di quattro bit-plane (16 colori).

LO SPAZIO DI MEMORIA RICHIESTO DALLO SCHERMO

Dopo che sono state fissate le dimensioni dello schermo, il BASIC conserva uno spazio sufficiente di RAM per contenere i bit-plane. Per calcolare quanta memoria richiede ogni schermo, è sufficiente moltiplicare le tre dimensioni dello schermo per ottenere il numero di bit e dividere poi questo numero per otto, ricavando la quantità di byte:

altezza x larghezza x profondità ÷ 8 =
RAM approssimativa dello schermo in byte

Ad esempio, uno schermo di 320 per 200 per 5 utilizzerebbe circa 40.000 byte di memoria (320x200x5:8)

L'Amiga BASIC riserva la RAM per gli schermi nella memoria sistema (non nei 25.000 byte messi da parte dal BASIC per memorizzare i programmi). Quando si creano gli schermi, bisogna tener conto di quanta memoria si ha a disposizione e fare attenzione a non sovraccaricarla. Il metodo migliore per conservare spazio in memoria è limitare la profondità dello schermo se non si ha bisogno di troppi colori. Uno schermo 320 per 200 per 5 mette a disposizione 32 colori, poiché utilizza 5 bit-plane. Se si riduce il numero di colori a 4 (2 soli bit-plane) serviranno solo 16.000 byte di memoria invece di 40.000.

ESEMPI DI COMANDO SCREEN

Dopo aver deciso il modo di risoluzione, le dimensioni e il numero di ID, è molto facile creare uno schermo. Ad esempio, per attivare uno schermo nel modo 4, con un numero di ID di 1, che misuri 640 per 400 pixel e abbia 2 bit-plane, immettete:

```
SCREEN 1, 640, 400, 2, 4
```

Per vedere come realizzare allo stesso tempo vari schermi con risoluzioni diverse, scrivete e fate girare questo breve programma:

```
SCREEN 1, 320, 200, 1, 1
SCREEN 2, 640, 200, 1, 2
SCREEN 3, 320, 400, 1, 3
SCREEN 4, 640, 400, 1, 4
```

Questo programma crea quattro schermi a tutto quadro, uno per ogni risoluzione, ognuno con una profondità di un solo bit-plane (per conservare spazio in memoria). Non sarà possibile vedere immediatamente gli schermi, in quanto essi scompaiono dietro lo schermo del Workbench appena il programma smette di girare. Per far apparire gli schermi sul monitor bisogna prima cambiare le dimensioni delle finestre Output e List e farle scorrere sul fondo dello schermo del Workbench in modo da vedere la linea d'intestazione del Workbench stesso. Poi, per vedere lo schermo dietro al Workbench, fate scorrere quest'ultimo verso il basso puntando la linea d'intestazione. Apparirà la linea d'intestazione dello schermo numero 4. Spostando questo verso il basso, apparirà lo schermo 3. Lo scorrimento dello schermo 3 rivelerà a sua volta lo schermo 2 e infine, spostato anche il 2, vedrete il numero 1. A questo punto sono visualizzati sul monitor cinque schermi contemporaneamente, come illustrato dalla Figura 4-4. È possibile usare anche i gadget per mettere uno schermo davanti o dietro agli altri.

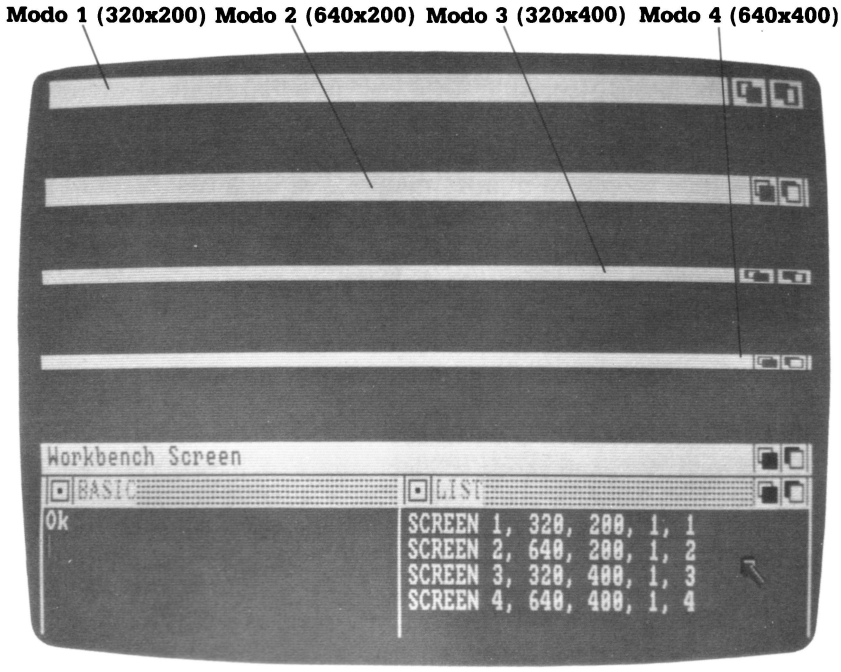


Figura 4-4.

Visualizzazione contemporanea di cinque schermi sul monitor.

Nella Figura 4-4 si può notare come le dimensioni delle linee d'intestazione e dei gadget cambino da schermo a schermo, in base alla risoluzione adottata. Inoltre gli schermi sono disposti nell'ordine in cui sono stati creati: ogni nuovo schermo attivato appare davanti a quelli creati in precedenza.

IL COMANDO SCREEN CLOSE

Dopo aver finito di utilizzare uno schermo è bene chiuderlo per farlo sparire dal monitor e per liberare la RAM che l'Amiga utilizza per memorizzarlo. Questa funzione è svolta dal comando SCREEN CLOSE, che ha la seguente sintassi:

```
SCREEN CLOSE numero di ID dello schermo
```

Il numero di ID specifica quale schermo si ha intenzione di chiudere. Ad esempio, per chiudere lo schermo 3 si scriverà:

```
SCREEN CLOSE 3
```

Per riportare lo schermo alle condizioni originali, modificate con l'immissione del programma precedente che ha creato una serie di schermi, è sufficiente riportare la linea d'intestazione dello schermo del Workbench nella parte superiore del monitor, selezionare la finestra Output premendo il tasto del mouse e inserire poi questi quattro comandi:

```
SCREEN CLOSE 4
SCREEN CLOSE 3
SCREEN CLOSE 2
SCREEN CLOSE 1
```

Si noterà che il tremolio dell'indirizzamento interlace scompare non appena si chiudono gli schermi 3 e 4.

IL COMANDO WINDOW

Quando si crea uno schermo, si stabilisce la risoluzione e il numero di colori con cui si vuole lavorare. Per creare sullo schermo un'area in cui inserire grafica e testo, bisogna aprire una finestra con il comando WINDOW. Si possono creare tante finestre quante ne può contenere la memoria dell'Amiga, e ogni finestra può avere posizione, dimensioni e caratteristiche diverse. Si possono creare finestre sullo schermo del Workbench

o su qualsiasi altro schermo attivato col comando SCREEN.
La sintassi del comando WINDOW è la seguente:

WINDOW *numero di ID della finestra, titolo, indirizzi angoli opposti, numero caratteristiche della finestra, numero di ID dello schermo*

Tutti i parametri devono apparire in quest'ordine, separati da virgole. Il numero di ID della finestra deve essere inserito in tutti i comandi WINDOW. Le altre indicazioni sono invece opzionali, come vedremo in seguito, poiché non sono necessarie per tutte le finestre. Comunque, se una indicazione non viene inserita, ma è seguita da altre, è necessario rimpiazzarla con una virgola.

IL NUMERO DI ID DELLA FINESTRA

Per identificare e non perdere il conto delle finestre, è necessario assegnare un numero di ID a ogni finestra che si è creata. Il numero può essere un intero qualsiasi compreso fra 1 e 255. Il numero di ID 1 corrisponde alla finestra Output del BASIC; così, se si vuole creare una nuova finestra senza alterare quella di Output, è necessario numerare le finestre partendo dal numero 2.

IL TITOLO DELLA FINESTRA

Il parametro titolo nel comando WINDOW offre la possibilità di inserire un titolo che compaia nella linea d'intestazione della finestra. Si può utilizzare una stringa qualsiasi, ad esempio una stringa fra virgolette tipo "Ehilà!", oppure una variabile stringa come TITOLO$, definita precedentemente nel programma. È buona regola terminare sempre la stringa con uno spazio vuoto per far apparire il titolo più chiaro e meno compresso nell'intestazione. Se non si desidera alcun titolo, ma si vogliono specificare altri parametri che lo seguono, è necessario inserire una virgola al posto della stringa. L'intestazione della finestra risulterà così priva di titolo.

GLI INDIRIZZI DEGLI ANGOLI OPPOSTI

Per stabilire le dimensioni e la posizione di una finestra bisogna specificare la locazione di due dei suoi angoli opposti (superiore sinistro e inferiore destro). A questo scopo è necessario ricorrere all'indirizzamento dei pixel. I pixel sono disposti sullo schermo come in un reticolo che utilizza linee e colonne numerate a partire dall'angolo superiore sinistro dello schermo.

La prima linea di pixel è la linea 0, poi seguono le linee 1, 2, 3, 4 e così via, fino al fondo dello schermo. La prima colonna di pixel sulla sinistra è la colonna 0, seguita dalle colonne 1, 2, 3, 4 e così via, fino al margine destro dello schermo. La Figura 4-5 mostra come funziona questo sistema di numerazione.

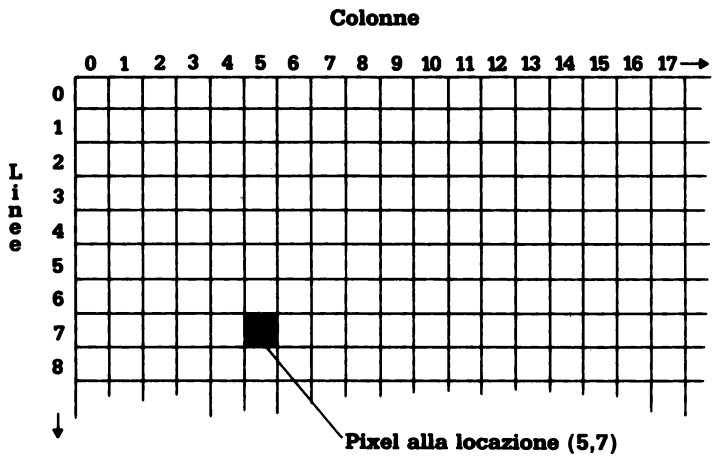


Figura 4-5.

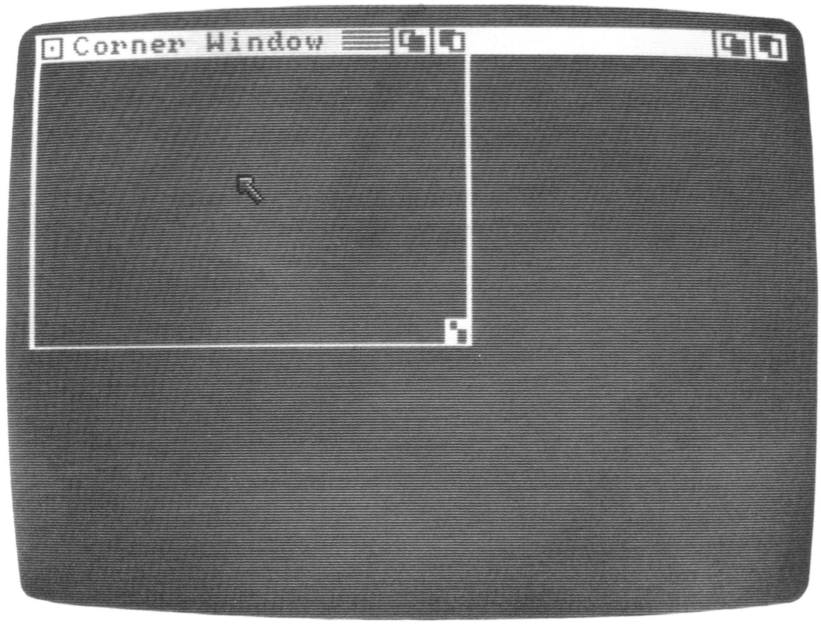
Il sistema di indirizzamento dei pixel dell'Amiga BASIC.

Per posizionare un pixel si usa un indirizzo composto dai numeri della colonna e della linea in cui si trova il pixel in questione. Ad esempio, un pixel con indirizzo 5,7 è situato 6 colonne a destra e 8 righe più in basso rispetto all'angolo superiore sinistro dello schermo. Dal momento che la numerazione parte dallo zero, la linea e la colonna effettive saranno sempre maggiori di uno rispetto ai numeri indicati nell'indirizzo.

Nell'Amiga BASIC gli indirizzi dei pixel sono sempre indicati tra parentesi, con il numero della colonna seguito da una virgola e poi dal numero della linea. In un programma BASIC l'indirizzo del pixel indicato in precedenza apparirà come (5,7). Così, una finestra che occupa la parte superiore sinistra di uno schermo 320 per 200 verrà indicata come (0,0) - (159,99). Questi due indirizzi corrispondono rispettivamente all'angolo superiore sinistro e al centro dello schermo. La finestra che creano è simile a quella della Figura 4-6 (riportata nella pagina successiva).

Figura 4-6.

Una finestra creata nella parte superiore dello schermo.



Inserendo gli indirizzi degli angoli opposti, è bene assicurarsi che siano entro i limiti dello schermo. I limiti sono determinati dalla risoluzione e dalle dimensioni stabilite col comando SCREEN. In uno schermo di 640 per 400 pixel non si potrà creare una finestra con angoli opposti in (10,20) - (300,450), perché il secondo indirizzo oltrepassa il fondo dello schermo.

Se volete aprire una finestra che occupi l'intero schermo, è sufficiente non inserire gli indirizzi degli angoli opposti, sostituendoli con una virgola.

CARATTERISTICHE DELLE FINESTRE

Come avrete già scoperto utilizzando diversi programmi per l'Amiga, le finestre hanno dimensioni e caratteristiche diverse. Alcune possono scorrere sullo schermo mentre altre sono fisse. Alcune dispongono di ogni tipo di gadget, altre non ne hanno neanche uno.

Il comando WINDOW offre la possibilità di personalizzare la finestra al momento della sua creazione. È possibile selezionare le caratteristiche nello stesso modo in cui si scelgono articoli da un catalogo. Ogni caratteristica è associata a un valore numerico. Dopo aver deciso quali saranno le particolarità della propria finestra, si sommano tra loro i valori delle

caratteristiche selezionate. La cifra risultante è il numero delle caratteristiche della finestra e indica al comando WINDOW quali particolarità sono state prescelte. La lista seguente indica le caratteristiche che si possono specificare e i loro valori corrispondenti:

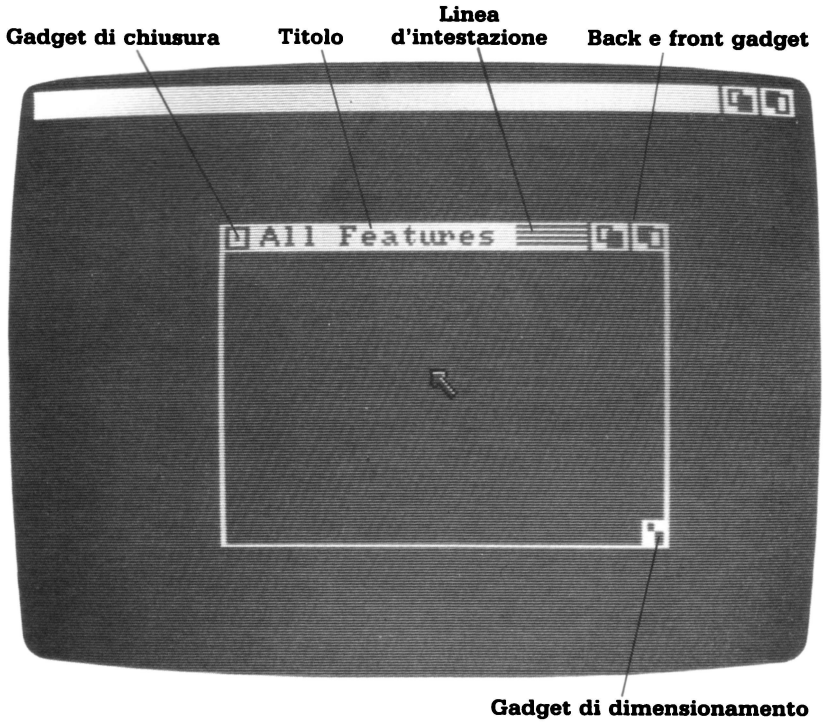
- **Gadget di dimensionamento (1)** - È il gadget che appare nell'angolo inferiore destro della finestra. Muovendolo col mouse consente di modificare le dimensioni della finestra. Senza questo dispositivo, una finestra non può essere modificata dall'utente (caratteristica a volte utile).
- **Scorrimento della finestra (2)** - Puntando la linea d'intestazione col mouse, è possibile far scorrere la finestra sullo schermo. Una finestra senza questa caratteristica non può essere spostata dalla sua posizione originale.
- **Back e front gadget (4)** - Sono i gadget posti nell'angolo superiore destro della finestra. Possono essere utilizzati per portare una finestra davanti (front gadget) o dietro (back gadget) alle altre finestre dello schermo.
- **Gadget di chiusura (8)** - È il gadget che appare nell'angolo superiore sinistro della finestra. Selezionare questo gadget col mouse equivale a chiudere la finestra, facendola sparire dallo schermo. Senza gadget di chiusura la finestra non può essere rimossa utilizzando il mouse.
- **Window refreshing (16)** - Questo dispositivo consente la rivisualizzazione (refreshing) del contenuto di una finestra che sia stata coperta da un'altra finestra o sia stata ridotta di dimensioni e poi nuovamente allargata. Senza "refreshing", sarà cancellato definitivamente qualsiasi punto della finestra che venga coperto da un'altra finestra o che venga tagliato quando la finestra viene rimpicciolita.

Nella pagina successiva, la Figura 4-7 mostra una finestra con tutte le caratteristiche che su di essa si possono vedere (naturalmente non può mostrarvi la finestra che si muove o che viene modificata).

Facciamo un esempio di come si determinano le caratteristiche di una finestra: se volete una finestra con tutti i tipi di gadget, sommate i valori delle caratteristiche che vi interessano (1, 4 e 8) e otterrete 13. Immettendo il risultato come numero delle caratteristiche della finestra nel parametro del comando WINDOW, saranno selezionate le particolarità prescelte. Se non viene indicato alcun numero e si scrive semplicemente una virgola, il comando WINDOW presume che siano state richieste tutte le caratteristiche.

Figura 4-7.

Una finestra dell'Amiga BASIC completa di ogni caratteristica.



Non tutte queste caratteristiche sono facilmente disponibili. Alcune particolarità richiedono un grande spazio di memoria ed è quindi preferibile non specificarle se non sono strettamente necessarie. Il "refreshing" richiede parecchia RAM per memorizzare il contenuto della finestra; naturalmente la quantità di memoria aumenta all'aumentare delle dimensioni della finestra. Allo stesso modo, una finestra con un gadget di dimensionamento, indipendentemente dalla grandezza, richiede la medesima quantità di RAM di una finestra tanto grande quanto lo schermo, poiché una finestra che abbia quel gadget può essere trasformata in qualsiasi momento in una finestra a tutto schermo.

Una caratteristica che non viene indicata esplicitamente nell'elenco precedente è la linea d'intestazione. Se selezionate i back e front gadget, il gadget di chiusura o lo scorrimento della finestra, oppure assegnate un nome alla finestra, l'intestazione sarà automaticamente inserita. Se non scegliete alcuna di queste caratteristiche l'intestazione della finestra non sarà necessaria e verrà quindi omessa.

IL NUMERO DI ID DELLO SCHERMO

L'ultimo parametro da specificare nel comando WINDOW è il numero di ID dello schermo. Esso indica lo schermo su cui si vuole aprire la finestra ed è importante perché la finestra utilizza la risoluzione e la palette dello schermo in cui si trova. Il numero di ID può essere -1, 1, 2, 3 o 4 e corrisponde al numero di ID fissato al momento della creazione dello schermo con il comando SCREEN. Non si può specificare il numero di uno schermo che non esiste; se chiedete al comando WINDOW di creare una finestra sullo schermo 2 e lo schermo in questione non esiste, il BASIC vi risponderà con un messaggio di errore.

Volendo creare una finestra sullo schermo del Workbench è necessario specificare il numero di ID -1. Se non si specifica alcun numero di schermo, il BASIC sceglierà automaticamente lo schermo del Workbench, ed è lì che creerà la finestra.

ESEMPI DI COMANDO WINDOW

Per avere un'idea di come vengano create le finestre, pensate di creare una finestra chiamata "Dirty Window" sullo schermo 1, che ha due bit-plane e modo di risoluzione 1, con dimensioni di 320 per 200 pixel. La finestra creata sarà la numero 2 e apparirà nel mezzo dello schermo, dotata di tutte le caratteristiche escluso il "refreshing". Il comando WINDOW che consente di aprire questa finestra è:

```
WINDOW 2, ''Dirty Window '', (79,49) - (239,149), 15, 1
```

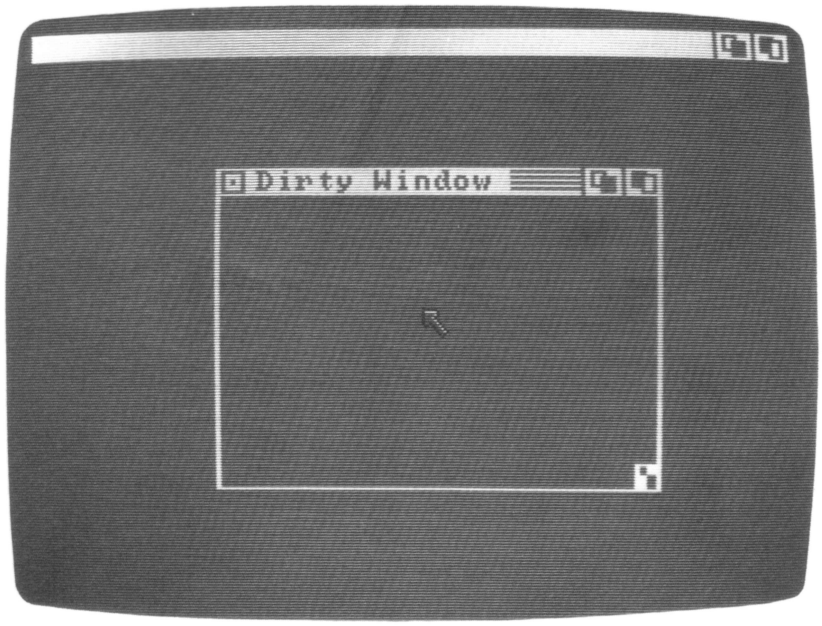
Se volete verificare, scrivete e fate girare il seguente programma:

```
SCREEN 1, 320, 200, 2, 1
WINDOW 2, ''Dirty Window '', (79,49) - (239,149), 15, 1
```

Lo schermo creato da questo programma apparirà come quello della Figura 4-8. È possibile spostare la finestra con il puntatore del mouse e usare i vari gadget. Per tornare allo schermo del Workbench, selezionate **Show List** dal menu **Windows**. Riappariranno le finestre Output e List.

Figura 4-8.

La finestra realizzata col programma "Dirty Window".



Come abbiamo già detto, l'unica indicazione che deve necessariamente comparire nel comando WINDOW è il numero di ID della finestra. Il comando

```
WINDOW 2
```

crea la finestra numero 2, che ha tutte le caratteristiche delle finestre, è priva di titolo e occupa tutto lo schermo del Workbench.

Se l'utente desidera inserire un solo parametro alla fine del comando senza immetterne altri, bisogna inserire delle virgole al posto dei parametri precedenti, altrimenti si commette un errore di sintassi. Per aprire la finestra numero 2 nello schermo numero 1 senza ulteriori indicazioni, è necessario utilizzare il comando:

```
WINDOW 2, , , , 1
```

LA FINESTRA OUTPUT

Anche se è stata aperta più di una finestra, l'Amiga BASIC disporrà grafica e testo in una sola finestra per volta. Se non fosse così, il BASIC non saprebbe quale finestra utilizzare per

visualizzare i risultati dei comandi grafici e di stampa. La finestra in cui vengono esposti disegni e testi è chiamata finestra output.

Quando si creano finestre con il comando WINDOW, l'ultima finestra realizzata diventa automaticamente la finestra output. Tutte le creazioni dei comandi grafici e i testi dei comandi di stampa appariranno in questa finestra.

È possibile utilizzare il comando WINDOW per "ricreare" una finestra, portandola davanti alle altre e facendola diventare la finestra output. Ad esempio, il breve programma riportato qui di seguito crea due finestre, 2 e 3, e poi ricrea la 2, riportandola davanti alla 3 nella stessa posizione e con le stesse dimensioni originali, facendola diventare la finestra output. Il comando PRINT alla fine del programma permette di verificare qual è la finestra output.

```
WINDOW 2, ''Finestra 2 '', (0,0) - (250,100)
WINDOW 3, ''Finestra 3 '', (200,0) - (400,100)
WINDOW 2
PRINT ''Questa e' la finestra output.''
```

Notate che la linea d'intestazione della finestra output presenta una luminosità attenuata. Solo l'intestazione della finestra input (la finestra in cui posizionate il puntatore e premete il tasto Select) è molto luminosa. Le finestre input e output non hanno niente a che vedere l'una con l'altra.

Notate anche che la terza riga del programma ricrea la finestra 2 senza cambiare le dimensioni e la posizione originali. Questo perché quando si usa un comando WINDOW senza specificazioni per una finestra che già esiste, il BASIC utilizza la prima serie di opzioni scelte per quella finestra nel primo comando WINDOW.

IL COMANDO WINDOW OUTPUT

Se volete trasformare in finestra output una finestra già creata, ma non volete spostarla davanti alle altre, potete utilizzare il comando WINDOW OUTPUT. La sintassi è la seguente:

```
WINDOW OUTPUT numero di ID della finestra
```

Il numero di ID, naturalmente, è quello della finestra che volete trasformare in finestra output. Dopo che il BASIC ha eseguito il comando WINDOW OUTPUT. la finestra in questione

visualizza i disegni e i testi senza spostarsi davanti alle altre. Sostituendo le due linee riportate qui di seguito all'ultima linea del programma precedente, la finestra 3 diventerà la finestra output pur rimanendo dietro la 2. Il comando PRINT stamperà quindi un messaggio nella finestra 3.

```
WINDOW OUTPUT 3
PRINT ''Ciao studioso di finestre!''
```

IL COMANDO WINDOW CLOSE

Avendo più finestre su uno schermo, noterete che ci vuole molto più tempo a spostare e modificare le finestre con il mouse, perché l'Amiga è rallentato dal numero di finestre che deve gestire. È quindi importante chiudere tutte le finestre che non si utilizzano per liberare la RAM e permettere all'Amiga di lavorare più velocemente.

Esistono due modi per chiudere una finestra. Se ha un gadget di chiusura, è sufficiente selezionarlo col mouse e la finestra scomparirà. Volendo chiudere la finestra dall'interno di un programma BASIC, si può utilizzare il comando WINDOW CLOSE, con la seguente sintassi:

```
WINDOW CLOSE numero di ID della finestra
```

Il numero di ID specifica quale finestra si vuole chiudere. Ad esempio, questo comando chiude la finestra numero 2:

```
WINDOW CLOSE 2
```

PALETTE E REGISTRI DI COLORE

Come ricorderete, ogni schermo ha la sua palette con la sua gamma di colori che utilizza per dipingere lo sfondo, i bordi delle finestre, le opzioni dei menu e qualsiasi elemento grafico o testo riprodotto sullo schermo. L'Amiga utilizza 32 locazioni diverse all'interno della RAM, chiamate registri di colore, per memorizzare le varie tonalità. I registri di colore sono numerati da 0 a 31, ed esiste una serie di 32 registri per ogni singolo schermo.

Quando si accende l'Amiga, i registri di colore sono già occupati dai colori di default (cioè quelli preselezionati), che l'utente può utilizzare qualora rispondano alle sue esigenze. Se volete vedere questi colori, fate girare il seguente programma:

```
SCREEN 1, 320, 200, 5, 1
WINDOW 2, ''Registri colore '', , , 1
PSET(0,190)
FOR i = 0 TO 31
    LINE STEP(0,-180) - STEP(9,180), i, BF
NEXT i
```

Questo programma disegna una serie di righe verticali colorate (parleremo nel prossimo capitolo dei comandi PSET e LINE, finora sconosciuti). Le righe di colori mostrano, da sinistra a destra, il contenuto dei registri di colore in ordine numerico da 0 a 31. Se vedete solo 31 colori è perché la riga all'estrema sinistra dello schermo è dello stesso colore dello sfondo.

L'Amiga segue regole ben precise per utilizzare i registri di colore. Il registro 0 è sempre il colore di sfondo, mentre il registro 1 è il colore di primo piano, utilizzato dall'Amiga per stampare i testi, i contorni delle finestre e altre immagini che appaiono nel Workbench. I registri 2 e 3 contengono i colori utilizzati per la colorazione inversa delle opzioni selezionate all'interno di un menu. I registri 17 e 20 contengono infine le tinte utilizzate per la freccetta del mouse; essa non fa parte dello schermo, pertanto può utilizzare sempre questi due registri anche se i loro colori non sono compresi nella palette dello schermo. Utilizzando il menu Preferences per modificare i colori del Workbench e del puntatore del mouse, si cambia automaticamente anche il contenuto dei registri di colore compresi fra 0 e 3, e 17 e 20.

Non è possibile utilizzare sempre tutti i registri in ogni schermo; bisogna limitarsi ai colori consentiti dal numero di bit-plane. La seguente tabella indica quali registri di colore vengono utilizzati a seconda dei bit-plane che formano lo schermo.

Numero di bit-plane	Registri di colore utilizzati
1	0 - 1
2	0 - 3
3	0 - 7
4	0 - 15
5	0 - 31

IL COMANDO COLOR

La maggior parte dei comandi grafici dell'Amiga BASIC crea immagini utilizzando per il primo piano il colore del registro 1. Lo stesso avviene per la stampa di testi. Alcuni comandi grafici utilizzano anche il colore dello sfondo, contenuto nel registro 0. È possibile cambiare entrambi i colori con altri disponibili per mezzo del comando COLOR, immesso con la seguente sintassi:

COLOR *numero di registro del colore di primo piano, numero di registro del colore di sfondo*

Per selezionare i due colori è sufficiente specificare i numeri dei registri che si vogliono utilizzare, separandoli con una virgola. Ad esempio, il comando che segue richiede il colore del registro 2 per il primo piano e quello del registro 3 per lo sfondo:

```
COLOR 2, 3
```

Tralasciando la virgola e il secondo numero, il comando COLOR modifica soltanto il colore del primo piano e lascia inalterato quello dello sfondo. Inserendo solo la virgola seguita da un numero, viene modificato solo il colore di sfondo ma non quello di primo piano. Se in un programma non viene utilizzato il comando COLOR, il BASIC usa automaticamente il colore del registro 1 per il primo piano e quello del registro 0 per lo sfondo. Se invece l'utente fissa due colori diversi, il BASIC li utilizza con qualsiasi comando grafico si presenti da quel momento in poi. Utilizzando COLOR fra un comando grafico e l'altro, si possono selezionare nuovi colori che si adattino di volta in volta alle immagini visualizzate sullo schermo.

Dopo aver cambiato il colore di primo piano, qualsiasi comando che stampi testi o figure, inserito dopo il comando COLOR, utilizzerà il nuovo colore, dal momento che i comandi BASIC che creano le immagini utilizzano il colore del primo piano (come vedremo nel prossimo capitolo). Per quanto riguarda lo sfondo, l'effetto è meno evidente. Non si vedrà alcun mutamento nel colore dello sfondo, a meno che non si utilizzi un comando che si serva di tale colore per svolgere la sua funzione. È importante ricordare che i cambiamenti di colore dello sfondo e del primo piano modificano il risultato

dei comandi grafici che seguono il comando COLOR, mentre tutto ciò che è stato creato fino a quel momento rimarrà sullo schermo invariato, a meno che non lo si ricrei nuovamente dopo aver immesso il nuovo comando COLOR.

Scegliendo i colori dello sfondo e del primo piano attraverso il comando COLOR, bisogna limitarsi ai colori contenuti nei registri inclusi nella palette dello schermo. Ad esempio, se si sta disegnando su uno schermo con due soli bit-plane, si potrà scegliere soltanto fra quattro colori. Se i colori dei quattro registri non soddisfano le vostre esigenze, è possibile cambiarli. Il comando PALETTE permette di sostituire il contenuto di qualsiasi registro con uno dei 4096 colori disponibili.

IL COMANDO PALETTE

Il comando PALETTE ha la seguente sintassi:

PALETTE *numero del registro di colore, quantita' di rosso, quantita' di verde, quantita' di blu*

Il numero del registro di colore, un numero intero compreso fra 0 e 31, specifica il registro che si vuole modificare. Le quantità di rosso, verde e blu devono essere specificate in numeri decimali compresi fra 0 e 1; questi numeri indicano le percentuali di rosso, verde e blu che si vogliono mescolare per creare un nuovo colore. Per capire come fissare le percentuali necessarie, bisogna prima sapere come un registro memorizza il colore.

MEMORIZZAZIONE DEI COLORI NEI REGISTRI

Ogni registro memorizza il colore come combinazione delle tinte di base rosso, verde e blu. Per indicare al BASIC quale colore si intende memorizzare nel registro in questione, bisogna indicare l'intensità di ognuno dei colori fondamentali che si vuole utilizzare per creare un nuovo colore. Vi sono 16 livelli di intensità per ogni colore fondamentale, per un totale di 4096 combinazioni disponibili (16 x 16 x 16 = 4096 colori).

Invece di specificare tutti i numeri da 1 a 16 (ad esempio, 5 parti di rosso, 8 di verde e 16 di blu), l'Amiga BASIC richiede di specificare l'intensità di ogni colore con un numero decimale

compreso fra 0 e 1. Lo 0 sta a significare che il colore in questione non è presente, mentre l'1 indica che il colore è presente in tutta la sua intensità. La tabella della Figura 4-9 aiuta a stabilire i numeri decimali da usare nel comando PALETTE per indicare l'intensità di ogni colore fondamentale. Utilizzando questa tabella, si può definire un colore come quello descritto precedentemente (5 parti di rosso, 8 di verde e 16 di blu) e trasformare questi valori in cifre decimali, quelle che il BASIC richiede nel comando PALETTE (nel nostro caso .28, .47 e 1).

Figura 4-9.

I 16 livelli dell'intensità dei colori primari e i loro corrispondenti valori PALETTE.

Intensità del colore	Valore PALETTE
1	0.00
2	.09
3	.15
4	.22
5	.28
6	.34
7	.40
8	.47
9	.53
10	.59
11	.65
12	.72
13	.78
14	.84
15	.90
16	1.00

Per creare un nuovo colore è necessario regolare l'intensità dei tre colori fondamentali e inserire dopo il comando PALETTE il numero del registro che si vuole modificare, seguito dai valori della tabella. Ad esempio, il seguente comando cambia il colore del registro 0 (colore di sfondo) in una sfumatura di verde, fissando il rosso a intensità .34, il verde a .84 e il blu a 0 (il blu è completamente assente).

```
PALETTE 0, .34, .84, 0
```

CREAZIONE DI PALETTE DIVERSE PER SCHERMI DIVERSI

Come ricorderete, ogni schermo dell'Amiga ha la propria serie di registri di colore. Quando si creano nuovi schermi con il comando SCREEN, l'Amiga BASIC riporta il contenuto dei registri del Workbench nei registri dei nuovi schermi. Per creare una gamma di colori completamente nuova è necessario usare una serie di comandi PALETTE, immediatamente a seguire il comando SCREEN. Il comando PALETTE va ad alterare solo i registri di colore dell'ultimo schermo creato.

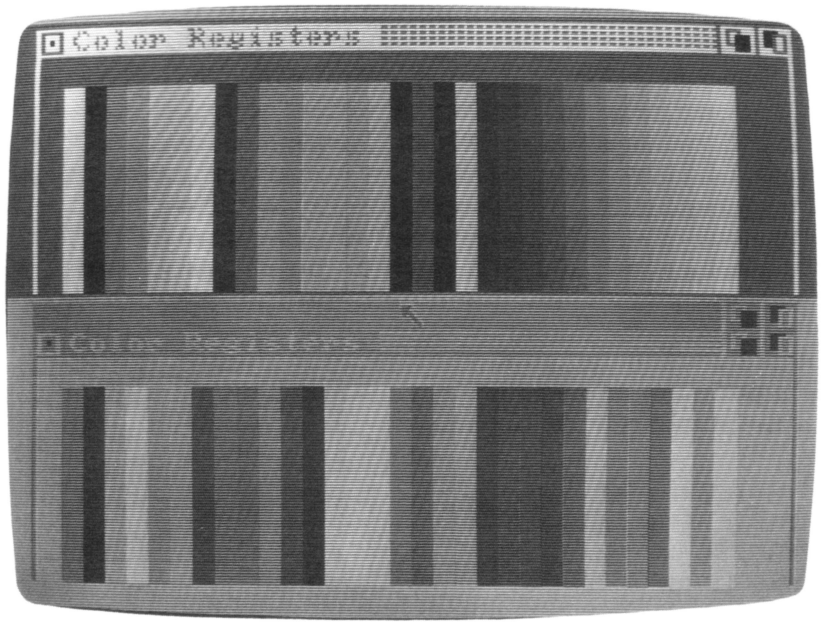
Ad esempio, il breve programma che segue visualizza le strisce verticali del programma visto poche pagine fa e crea poi una palette di colori completamente nuova, ritracciando infine le bande verticali con le tinte della nuova palette.

```
SCREEN 1, 320, 200, 5, 1
WINDOW 2, ''Color Registers '', , 15, 1
PSET(0,190)
FOR i = 0 TO 31
    LINE STEP(0,-180) - STEP(9,180), i, BF
NEXT i
SCREEN 2, 320, 200, 5, 1
WINDOW 3, ''Color Registers '', (0,10) - (297,185), 15, 2
FOR i = 0 TO 31
    PALETTE i, RND(1), RND(1), RND(1)
NEXT i
PSET(0,190)
FOR i = 0 TO 31
    LINE STEP(0,-180) - STEP(9,180), i, BF
NEXT i
```

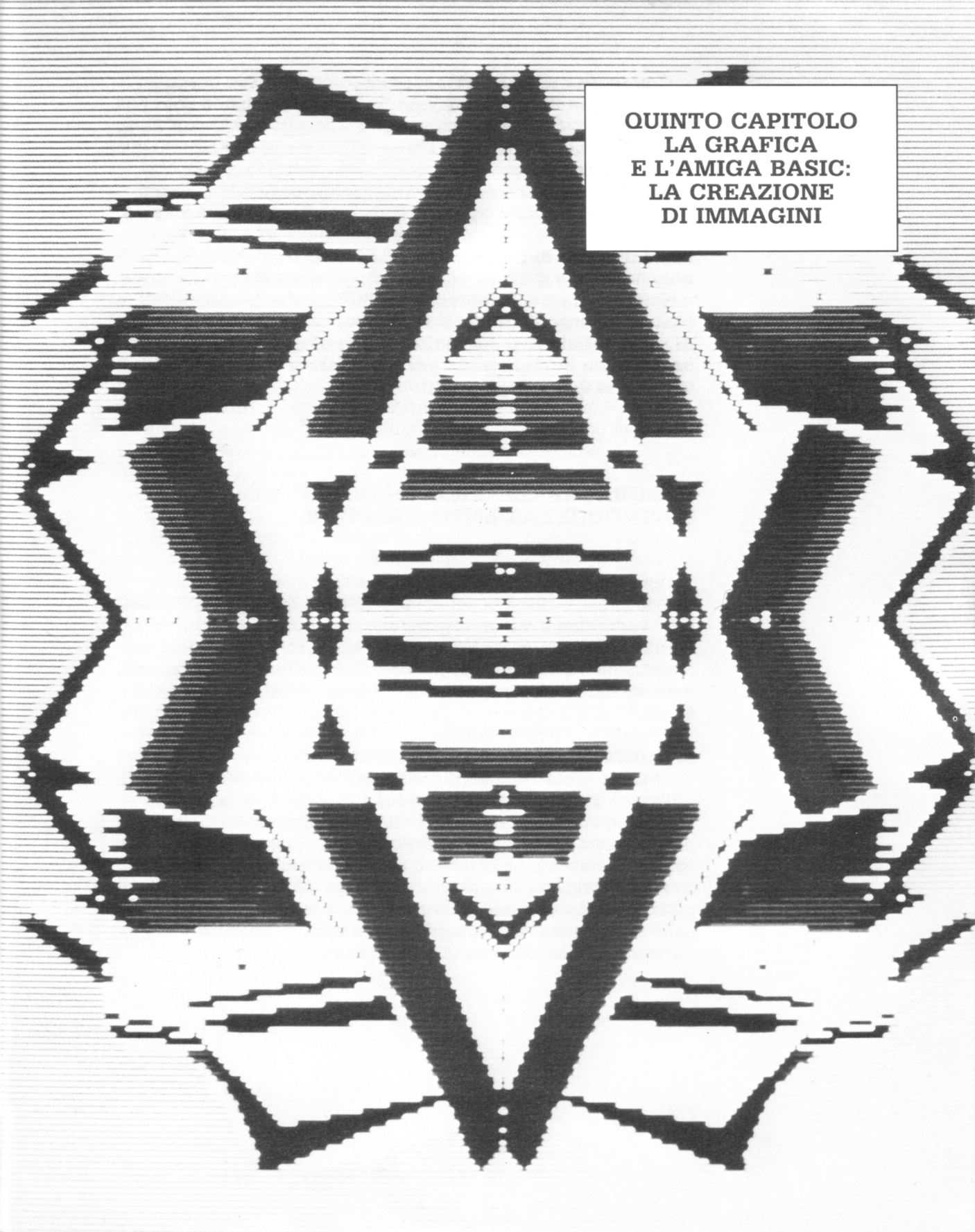
Il ciclo immediatamente successivo al comando WINDOW 3 utilizza il comando PALETTE per creare in modo casuale una gamma di colori per i registri del secondo schermo. Facendo girare il programma è possibile vedere i nuovi colori. Se fate scorrere verso il basso il nuovo schermo colorato a caso, vedrete apparire dietro di esso il primo schermo, formato da una gamma di colori totalmente diversi. Creando diversi schermi sul monitor, ognuno con la sua serie di colori, è possibile visualizzare contemporaneamente più di 32 colori, che costituiscono il limite massimo di uno schermo (vedere Figura 4-10).

Figura 4-10.

Due schermi diversi con due palette esclusive.



In questo capitolo è stata illustrata la tecnica di creazione degli schermi in modo da utilizzare il tipo di risoluzione desiderato. Abbiamo anche visto come si apre una finestra allo scopo di fissare le dimensioni dell'area in cui vengono poi visualizzate le immagini da voi create. Infine abbiamo visto come creare una palette completamente nuova che consenta di lavorare con i colori desiderati. L'unica cosa di cui non abbiamo ancora parlato è la creazione vera e propria delle immagini. Finora sono state chiarite le fasi preliminari, che possono essere considerate qualcosa di simile alla preparazione della tela e dei colori che ogni pittore esegue prima di iniziare a lavorare.
Il prossimo capitolo illustra come mettere i colori sulla tela, cioè come costruire un'immagine con i comandi grafici dell'Amiga BASIC.



QUINTO CAPITOLO LA GRAFICA E L'AMIGA BASIC: LA CREAZIONE DI IMMAGINI

Creare un nuovo schermo, aprire una finestra output e decidere quali colori utilizzare in una palette sono solo i primi passi verso la creazione di grafica in BASIC sull'Amiga. Una volta costruita l'area in cui intendete operare, potete incominciare a dar sfogo alla vostra fantasia, riempiendo le finestre con immagini di vostra ideazione.

In questo capitolo apprenderete i comandi dell'Amiga BASIC che consentono di disegnare figure semplici, cambiare il colore di un solo pixel o di un'intera finestra, nonché creare matrici grafiche diverse per disegnare linee e riempire figure. Imparerete anche un nuovo sistema di indirizzamento dei pixel e scoprirete come fissare i colori di primo piano e di sfondo. Alla fine di questo capitolo sarete sulla buona strada per diventare a tutti gli effetti un artista dell'Amiga BASIC.

IL CURSORE GRAFICO E L'INDIRIZZAMENTO DEI PIXEL

Ogni comando grafico dell'Amiga BASIC che utilizzerete in questo capitolo disegna servendosi di un cursore grafico invisibile. Per ordinare a ogni comando grafico dove iniziare e terminare la propria funzione, è necessario comprendere il funzionamento del cursore grafico e il modo in cui utilizzare gli indirizzi dei pixel per specificare un pixel di partenza e uno di arrivo.

IL CURSORE GRAFICO

Quando usate i comandi grafici dell'Amiga BASIC che utilizzano gli indirizzi dei pixel, sulla superficie della finestra si muove un cursore grafico invisibile della dimensione di un pixel. Questo cursore grafico si sposta sui pixel indicati dal comando grafico e ne cambia il colore. Se ricorrete a un comando semplice, che modifica soltanto un pixel, il cursore grafico si sposta sul pixel, ne muta il colore e rimane fermo nello stesso punto finché non viene mosso con un altro comando grafico. Se vi servite di un comando grafico più

complesso che utilizza due indirizzi. come ad esempio un comando che disegna una linea o un riquadro, il cursore grafico comincia a operare al primo indirizzo per terminare la sua funzione al secondo.

Nell'Amiga BASIC esistono due metodi per posizionare il cursore grafico su un pixel contenuto in una finestra, chiamati indirizzamento assoluto e indirizzamento relativo.

INDIRIZZAMENTO ASSOLUTO

L'indirizzamento assoluto è lo stesso sistema di indirizzamento usato per fissare gli angoli opposti di una finestra sullo schermo (descritto nel paragrafo dedicato al comando WINDOW nel capitolo precedente); l'unica differenza risiede nel fatto che in questo caso l'indirizzo specificato deve rispettare i limiti di una finestra e non di uno schermo. Nell'indirizzamento assoluto l'indirizzo deve essere specificato riferendosi sempre all'angolo superiore sinistro della finestra. I pixel sono numerati per colonne e linee, partendo dall'origine (0,0), situata nell'angolo superiore sinistro della finestra. Gli indirizzi sono racchiusi tra parentesi; il numero della colonna è separato dal numero della linea con una virgola. Ad esempio, (13,20) indica un indirizzo situato 14 colonne a destra e 21 linee più in basso rispetto all'angolo superiore sinistro (0,0) della finestra.

INDIRIZZAMENTO RELATIVO

L'indirizzamento relativo permette di indicare la posizione di un nuovo pixel in termini di distanza dal pixel precedentemente indirizzato. Ogni indirizzo relativo inizia con la parola chiave STEP, seguita dalla distanza in colonne e in righe, separate da una virgola e racchiuse tra parentesi. Ad esempio, STEP(5,8) indica un nuovo indirizzo situato 5 colonne a destra e 8 linee più in basso rispetto all'ultimo pixel indirizzato. Se volete utilizzare un indirizzo relativo per muovervi verso sinistra o verso l'alto rispetto all'ultimo pixel, dovete ricorrere ai numeri negativi. Ad esempio, STEP(-7,-9) indica un nuovo indirizzo 7 colonne a sinistra e 9 colonne sopra il vecchio indirizzo.

L'indirizzamento relativo vi fa risparmiare il tempo che impieghereste a calcolare un nuovo indirizzo assoluto. Ad esempio, se utilizzate un indirizzamento assoluto e iniziate con un pixel in locazione (10,5), e poi volete fissare un pixel 10

colonne a destra e 20 righe più in basso, dovete aggiungere 10 al numero delle colonne e 20 al numero delle linee per ottenere così un nuovo indirizzo assoluto, cioè (20,25). Se usate l'indirizzamento relativo, potete fissare il secondo indirizzo usando STEP(10,20) per muovervi 10 colonne verso destra e 20 linee verso il basso. Il funzionamento di queste procedure è illustrato nella Figura 5-1.

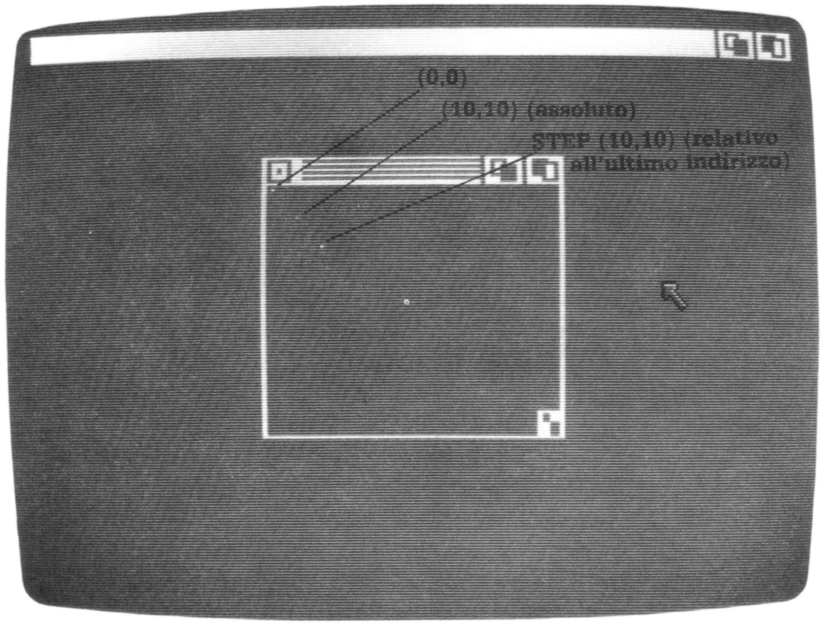


Figura 5-1.

L'indirizzamento relativo usato per specificare la locazione di un nuovo pixel.

L'indirizzamento relativo funziona particolarmente bene nei cicli FOR...NEXT, nei quali l'indirizzo cambia a ogni ciclo. Invece di calcolare nuovi indirizzi assoluti per ogni ciclo, un semplice indirizzamento relativo cambierà l'indirizzo di un valore pari all'incremento del ciclo a ogni iterazione. Ad esempio, il programma presentato nella Figura 5-2 utilizza il comando PSET (di cui parleremo tra poco) per fissare un punto sullo schermo situato 100 colonne a destra e 50 linee più in più basso rispetto all'origine (0,0), utilizzando un indirizzo assoluto. Poi utilizza un altro comando PSET all'interno di un ciclo FOR...NEXT per tracciare altri 10 punti, ognuno fissato facendo riferimento al punto immediatamente precedente.

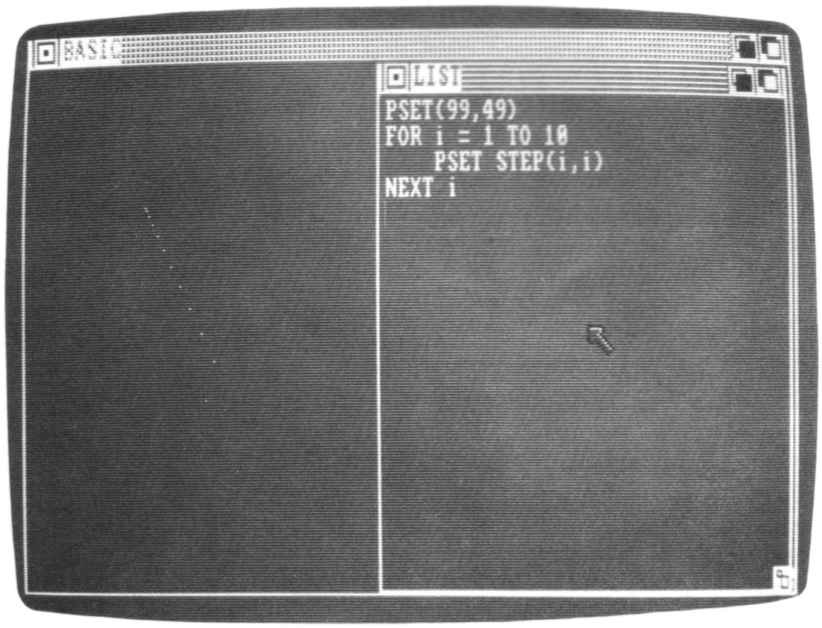


Figura 5-2.

Punti fissati con un ciclo FOR...NEXT e un indirizzamento relativo.

Ogni volta che utilizzate un indirizzo relativo, questo usa l'ultima posizione del cursore grafico come suo punto di riferimento. Questa posizione viene fissata normalmente dall'ultimo comando grafico. Ad esempio, se utilizzate un comando per tracciare una linea che finisce all'indirizzo (55,90) e poi utilizzate un comando grafico che usa l'indirizzo relativo STEP(10,10), il BASIC usa (55,90) come punto di riferimento per calcolare un nuovo indirizzo 10 colonne a destra e 10 righe più in basso. Il segreto per controllare il cursore grafico usando l'indirizzamento relativo sta nel conoscere sempre la locazione del cursore prima di usare un indirizzo relativo. In caso contrario potreste ottenere risultati imprevedibili. Se aveste dei dubbi, è preferibile usare l'indirizzamento assoluto.

I MARGINI DELLE FINESTRE

Quando si usano gli indirizzi dei pixel nei comandi grafici, bisogna assicurarsi di restare entro i margini della finestra output. Alcuni comandi, infatti, non funzionano se l'indirizzo specificato si trova all'esterno della finestra, mentre altri interrompono il programma e rispondono con un messaggio di errore.

Per creare una finestra con il comando WINDOW, l'utente deve stabilire le misure dell'interno della finestra, indicando gli indirizzi di due angoli opposti (per maggiori dettagli vedere il quarto capitolo). Se si crea una finestra con angoli opposti in (50,20) e (250,120), con linea d'intestazione e senza gadget di dimensionamento né front e back gadget, si avrà una finestra la cui superficie interna misura 201 pixel in larghezza (la distanza tra la colonna 50 e la colonna 250) e 101 pixel in altezza (la distanza dalla linea 20 alla linea 120 compresa). La finestra che appare sullo schermo è però più ampia di 201 per 101, in quanto include anche il margine e la linea d'intestazione che non vengono calcolati come superficie interna. La finestra comprende anche un margine interno dello spessore di un pixel sui bordi superiore e inferiore, e di due pixel sui margini laterali (vedere Figura 5-3). Essendo dello stesso colore dello sfondo, il margine interno non è visibile. È impossibile sfruttare questi pixel in quanto il margine interno "invisibile" è sempre presente e non è possibile disabilitarlo.

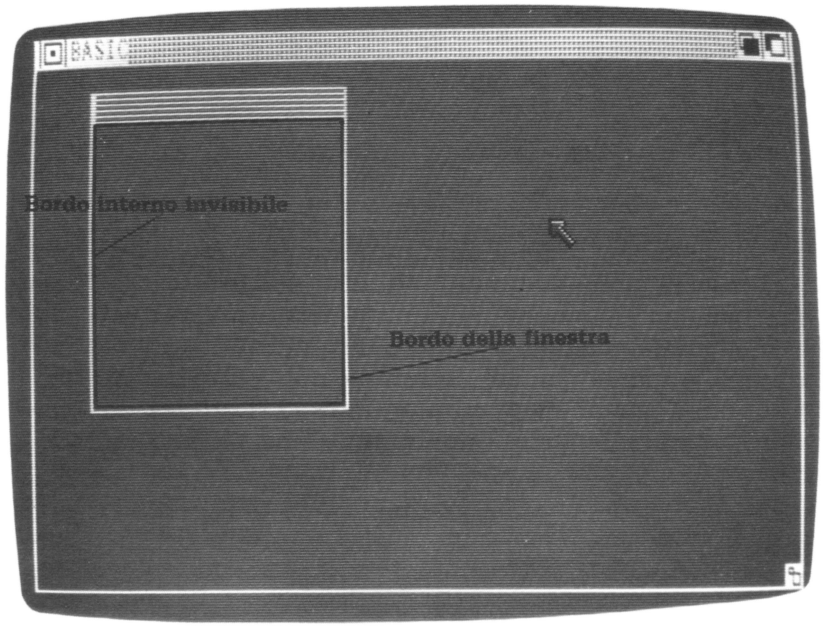


Figura 5-3.

L'area utilizzabile all'interno di una finestra con angoli opposti nelle locazioni (50,20) e (250,120).

Se si aggiunge un gadget di dimensionamento alla finestra, questa si allarga di 15 pixel per poterlo contenere. Non è però possibile disegnare nelle prime 10 nuove colonne a partire dal margine destro, quindi di fatto si guadagnano solo 5 colonne di spazio utilizzabile all'interno della finestra. Se si elimina la linea d'intestazione, la superficie interna della finestra ne occupa lo spazio, e si guadagnano così 10 nuove linee di pixel alla base della finestra. La Figura 5-4 presenta una finestra con gli stessi indirizzi della Figura 5-3, ma con un gadget di dimensionamento e senza linea d'intestazione. La superficie interna della finestra è di 206 colonne per 111 linee, 5 pixel più larga e 10 pixel più alta della stessa finestra con linea d'intestazione e senza gadget di dimensionamento.

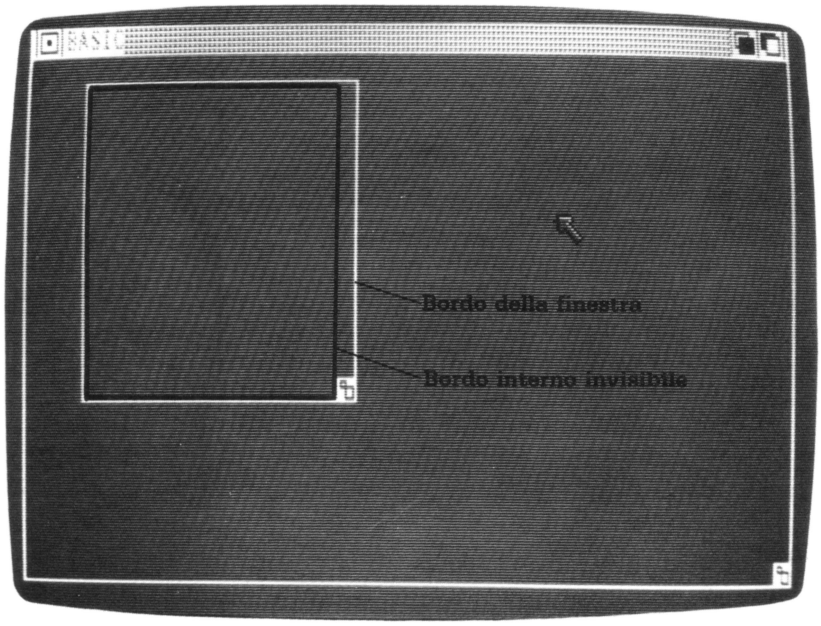


Figura 5-4.

L'area utilizzabile all'interno di una finestra con angoli opposti nelle locazioni (50,20) e (250,120), senza linea d'intestazione e con un gadget di dimensionamento.

Quando si usano comandi grafici in una finestra, gli indirizzi resteranno all'interno della finestra se saranno rispettati i limiti posti dagli indirizzi ad angoli opposti e dal margine invisibile. Un altro sistema per mantenere gli indirizzi entro i margini delle finestre consiste nel leggere le dimensioni dei margini con la funzione WINDOW( ), trattata nel prossimo capitolo.

IL COMANDO CLS

Prima di iniziare a creare una figura in una finestra, o qualora l'utente desideri cancellare immagini già presenti nella finestra, è possibile liberarsi dell'intero contenuto di una finestra usando il comando CLS. La sintassi è semplice:

```
CLS
```

CLS cancella tutto il quadro, riempiendo la finestra output del colore dello sfondo. Il programma seguente ne dimostra l'efficacia. Esso ricorre al comando COLOR per cambiare quattro volte il colore dello sfondo e poi usa il comando CLS per riempire lo schermo con ciascuno dei colori selezionati.

```
FOR i = 1 TO 3
    COLOR , i
    CLS
    FOR t = 1 TO 2000: NEXT t
NEXT i
COLOR , 0
CLS
```

IL COMANDO PSET

PSET è un comando molto semplice: cambia il colore di un singolo pixel della finestra con un colore specificato dall'utente. Combinando più comandi PSET è possibile creare immagini anche complesse. La sintassi è semplice:

PSET*(indirizzo pixel), codice-colore*

L'indirizzo del pixel può essere assoluto o relativo. Il codice-colore è facoltativo. Se viene specificato, va separato dall'indirizzo da una virgola; se viene tralasciato, PSET colorerà il pixel col colore di default. Per esempio, il comando

```
PSET(149,99)
```

colora il pixel nella colonna 150, linea 100, usando il codice-co lore di default. Se l'operatore specifica il codice, PSET usa il colore del registro relativo anziché utilizzare quello di default. Per esempio,

```
PSET(199,149) , 3
```

usa il registro di colore 3 per riempire il pixel alla colonna 200, linea 150.

Ovviamente il registro di colore specificato deve essere presente nello schermo su cui state lavorando. Ad esempio, se state usando uno schermo formato da 2 bit-plane, come lo schermo di default del Workbench, è possibile indicare solo registri di colore compresi fra 0 e 3, in quanto questi schermi contemplano solo quattro registri (o codici-colore). Inoltre, quando si indica un registro di colore, questo vale per un solo pixel. I pixel successivamente indicati con il comando PSET useranno il colore di default, a meno che venga specificato un altro codice-colore.

IL COMANDO PRESET

Il comando PRESET funziona quasi esattamente come il comando PSET. L'unica differenza risiede nel colore: se non specificate il codice-colore, il pixel apparirà col colore dello sfondo e non con quello di primo piano. La sintassi è la stessa di PSET:

```
PRESET (indirizzo pixel), codice-colore
```

Il codice-colore è opzionale. Per esempio, il comando

```
PRESET(14,74)
```

colora il pixel indirizzato con il colore di default dello sfondo. PRESET è molto utile per cancellare pixel precedentemente creati con PSET.

Se l'indirizzo è seguito da una virgola e da un codice-colore, allora PRESET colora il pixel con il colore specificato (proprio come il comando PSET). Il numero di registri di colore disponibili dipende anche in questo caso dal numero di bit-plane che formano lo schermo in cui è contenuta la finestra output.

INTRODUZIONE DI FORMULE NEI COMANDI PSET E PRESET PER LA CREAZIONE DI FIGURE

Con una buona dose di pazienza, le capacità matematiche di un buon contabile e i metodi di disegno di un fabbricante di tappeti, è possibile creare un'immagine anche complessa servendosi solo del comando PSET seguito da indirizzi e colori diversi per ogni pixel. In tal caso può essere utile un foglio di carta millimetrata per tracciare le figure ed elaborare gli indirizzi dei pixel e i colori.

Esiste però un metodo più semplice, che consiste nell'affidare parte del lavoro a un ciclo FOR...NEXT. Per esempio, questo programma consente di tracciare una linea diagonale sullo schermo:

```
FOR i = 1 TO 100
    PSET (i,i)
NEXT i
```

Se avete un po' di dimestichezza con la trigonometria potete usarne le funzioni per ottenere risultati di un certo livello. Potete, ad esempio, visualizzare l'andamento di una funzione trigonometrica, usando il valore della variabile di un ciclo FOR...NEXT come uno dei due indirizzi (il numero di colonna o di linea) e una funzione trigonometrica che fissi il valore dell'altra metà dell'indirizzo. Il programma seguente usa una funzione SIN (seno) per creare un'onda sinusoidale sullo schermo, come rappresentato nella Figura 5-5.

```
FOR i= 0 TO 600
    PSET(i, 90 + 80 * SIN(6.28318 / 600 * i))
NEXT i
Loop: GOTO Loop
```

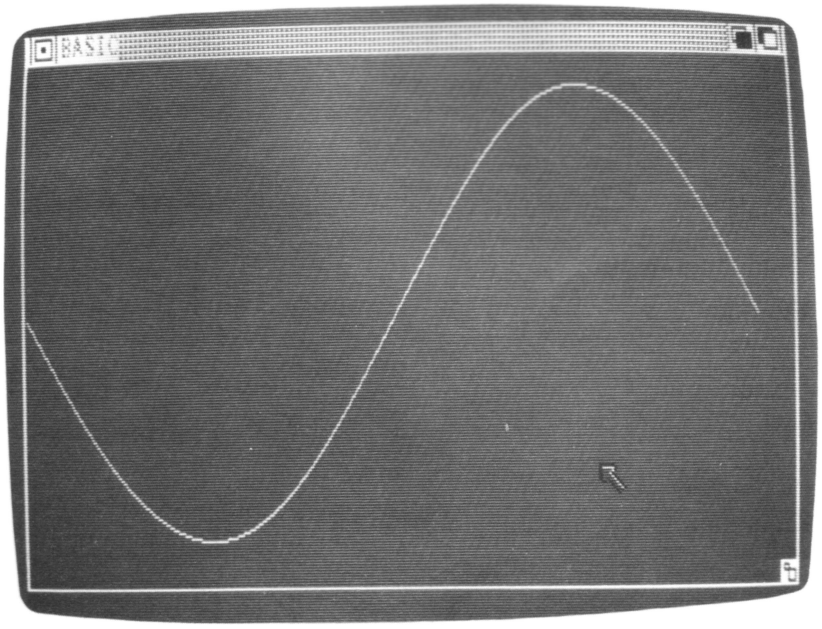


Figura 5-5.

Onda sinusoidale creata grazie all'abbinamento di una funzione trigonometrica al comando PSET.

Analizzando l'indirizzo della linea nel comando PSET, si noterà che esso utilizza il ciclo completo di un'onda sinusoidale (6.28318 ossia 6,28318, cioè *pi greco* moltiplicato per due), moltiplica il risultato per 80 in modo che l'onda tocchi sia il margine superiore che quello inferiore della finestra, e poi aggiunge 90 al valore ottenuto per centrare l'onda orizzontalmente. L'ultima linea di dati è in sostanza un "ciclo infinito", che continua a far girare il programma e tiene nascosta la finestra List, la quale potrebbe apparire coprendo la parte destra dell'onda visualizzata. Per interrompere il programma e visualizzare la finestra List, è necessario usare il tasto Menu del mouse e selezionare Stop dal menu Run.

Non è da tutti fare uso di soluzioni matematiche per creare immagini, ma se la cosa rientra nelle vostre capacità, potrete creare immagini davvero interessanti risparmiandovi il lavoro tedioso di dover tracciare ogni singolo pixel e mettendo finalmente in pratica tante formule teoriche.

IL COMANDO LINE

Finora sono stati descritti i comandi dell'Amiga BASIC che lavorano sui singoli pixel. Esiste però tutta una serie di comandi BASIC che permette di creare oggetti completi con un'unica istruzione. Uno di questi è il comando LINE. Usando questo comando in modi diversi, potrete realizzare tre tipi di figure geometriche: linee rette, quadrilateri vuoti e quadrilateri pieni (colorati col colore di primo piano). La sintassi del comando LINE è la seguente:

LINE *(indirizzo partenza) - (indirizzo arrivo) , codice-colore , opzioni box*

LINE deve presentare sia l'indirizzo di partenza, sia quello di arrivo, separati da un trattino. Gli indirizzamenti possono essere assoluti o relativi (questi ultimi sempre preceduti dalla parola chiave STEP). Il codice-colore e le opzioni box sono facoltative, e possono essere tralasciate. Se incluse, vanno separate da una virgola.

LA CREAZIONE DI LINEE

Per creare una linea è sufficiente fissare gli indirizzi del punto di partenza e di arrivo, in questo modo:

```
LINE(70,54) - (10,5)
```

Come potete vedere nella Figura 5-6, questo comando traccia una linea che congiunge il punto (70,54) al punto (10,5), utilizzando il colore di primo piano di default. Per usare qualsiasi altro colore è sufficiente specificare il codice-colore aggiungendo il numero desiderato al comando:

```
LINE(70,54) - (10,5) , 2
```

Questo comando traccia la stessa linea del comando precedente, ma la colora con il registro di colore 2 anziché col colore di default. Il cambiamento di colore vale solo per questa linea; quelle successive, se non viene specificato il codice-colore, torneranno a essere del colore di primo piano. Il numero di registri di colore disponibili dipende sempre dal numero di bit-plane che formano lo schermo e la finestra output.

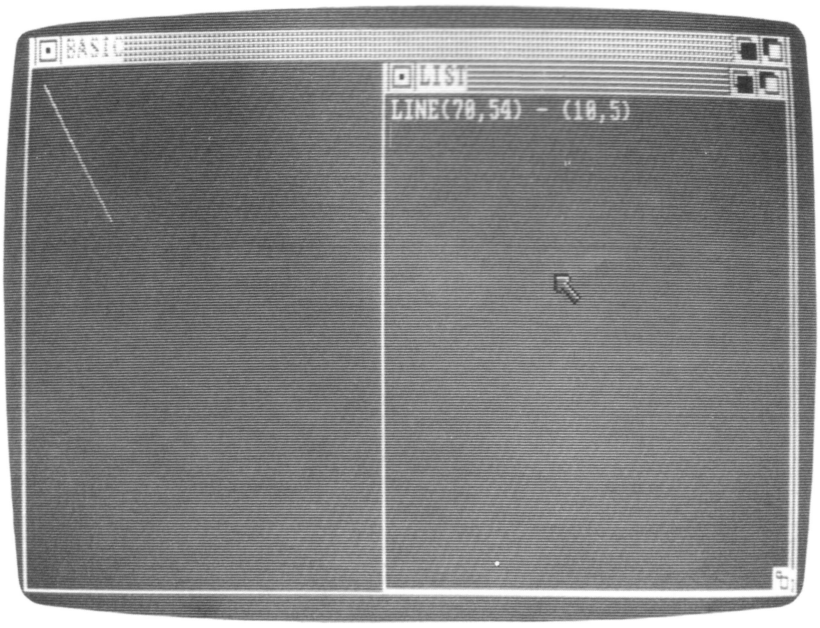


Figura 5-6.

Linea creata col comando LINE.

LA CREAZIONE DI QUADRILATERI

Per creare un riquadro con il comando LINE, bisogna specificare l'angolo superiore sinistro e quello inferiore destro, aggiungendo inoltre un'opzione dopo il codice-colore (se si desidera utilizzare il colore di primo piano di default, va comunque messa una virgola). Esistono due tipi di opzioni: B e BF. Con l'opzione B, LINE traccerà un riquadro vuoto invece che una linea, considerando i due indirizzi come gli angoli opposti della figura. Con l'opzione BF, invece, LINE disegnerà un riquadro colorato all'interno col colore di default. Ad esempio:

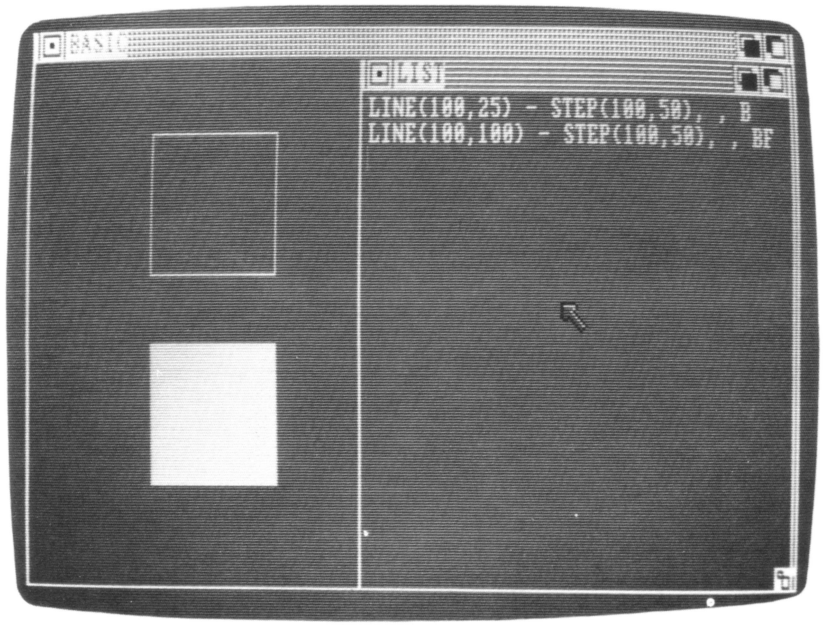
```
LINE(100,25) - STEP(100,50), , B
```

disegna un quadrilatero il cui angolo superiore sinistro si trova nella locazione (100,25) e il cui angolo inferiore destro è situato 100 colonne a destra e 50 linee più in basso (ricordate che STEP indica l'indirizzamento relativo). In questo esempio, il riquadro è vuoto, perché è stata usata l'opzione B, ed è disegnato nel colore di default, dato che non è stato specificato alcun codice-colore. Nella Figura 5-7 sono riportati i risultati, insieme a un riquadro pieno creato col seguente comando:

```
LINE(100,100) - STEP(100,50), , BF
```

Figura 5-7.

Il comando LINE seguito dall'opzione B crea un riquadro vuoto, mentre l'opzione BF dà origine ad un riquadro colorato col colore di primo piano.



IL COMANDO CIRCLE

Così come con le varianti del comando LINE si possono creare linee rette, angoli e riquadri, con il comando CIRCLE è possibile creare cerchi, ellissi e archi. CIRCLE ha la seguente sintassi:

CIRCLE *(indirizzo centro) , raggio , codice-colore , inizio arco , fine arco , aspetto*

Bisogna specificare l'indirizzo del centro della circonferenza e il valore del raggio. L'indirizzo centrale può essere assoluto o relativo. Il codice-colore, il punto iniziale e quello finale dell'arco e l'aspetto sono elementi facoltativi. Tutti i valori, qualora vengano inclusi, devono essere separati da virgole.

LA CREAZIONE DI CERCHI

Per creare un semplice cerchio è necessario fornire al comando CIRCLE un indirizzo per il centro della circonferenza e un valore per il raggio. Il raggio viene misurato come la distanza in pixel fra il centro e il perimetro della circonferenza. Per esempio, il comando per creare una circonferenza con centro in (150,100) e raggio di 50 pixel è:

```
CIRCLE(150,100) , 50
```

I risultati sono visibili nella Figura 5-8.

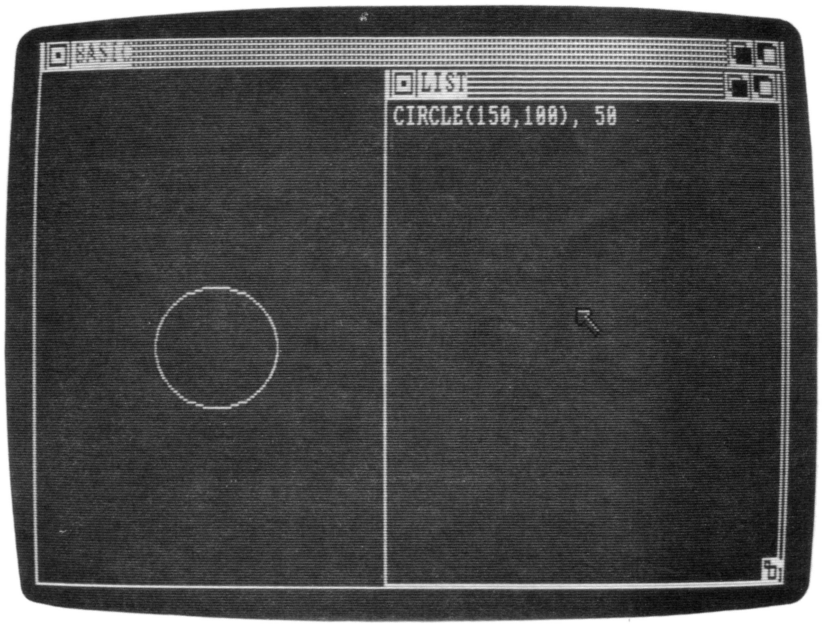


Figura 5-8.

Cerchio creato col comando CIRCLE.

CIRCLE crea generalmente un cerchio utilizzando il colore di primo piano di default. Se si desidera un colore differente, è sufficiente mettere una virgola dopo il valore del raggio e aggiungere il codice-colore prescelto. Il comando

```
CIRCLE(150,100) , 50 , 3
```

disegna lo stesso cerchio del comando precedente usando però il colore del registro 3.

LA CREAZIONE DI ARCHI

Per creare un arco è necessario aggiungere al comando CIRCLE un punto d'inizio e uno di arrivo, che determinino i due estremi dell'arco. Per specificare tali punti bisogna misurare la circonferenza in radianti e fornire a CIRCLE i due estremi dell'arco in termini di radianti. Immaginate il radiante come un'unità di misura che descriva il diametro di una circonferenza. Una circonferenza completa misura 6.28318 (6,28318) radianti; la metà 3.14159 (3,14159) radianti (valore corrispondente al *pi greco* matematico).

Pensando alla circonferenza come a un orologio, la misurazione in radianti per il comando CIRCLE inizia dalle ore 3, corrispondenti a zero radianti. La misurazione procede da questo punto in senso anti orario. Mezzogiorno corrisponde a

1.57079 (1,57079) radianti (*pi greco* diviso due), le nove equivalgono a 3.14159 (3,14159, *pi greco*), le sei a 4.71238 (4,71238, *pi greco* per 1,5) e un giro completo della circonferenza corrisponde a 6.28318 radianti (6,28318, *pi greco* per due).

Se il valore indicato per il punto di partenza è inferiore a quello del punto di arrivo, l'arco sarà tracciato in senso antiorario. Se il primo valore è invece superiore al secondo, l'arco sarà tracciato in senso orario.

Oltre a tracciare archi si possono anche aggiungere linee rette che hanno origine dal punto di partenza e/o dal punto di arrivo dell'arco e convergono nel centro della circonferenza, semplicemente rendendo negativo il valore del punto di partenza e/o di arrivo dell'arco. Il BASIC traccia l'arco come se i valori fossero positivi, poi disegna la linea che va dal punto indicato in numeri negativi al centro della circonferenza. Il comando seguente traccia un arco con una linea retta che parte dal punto d'arrivo dell'arco stesso, come rappresentato nella Figura 5-9.

```
CIRCLE(150,100),50 , , .75 , -5.6
```

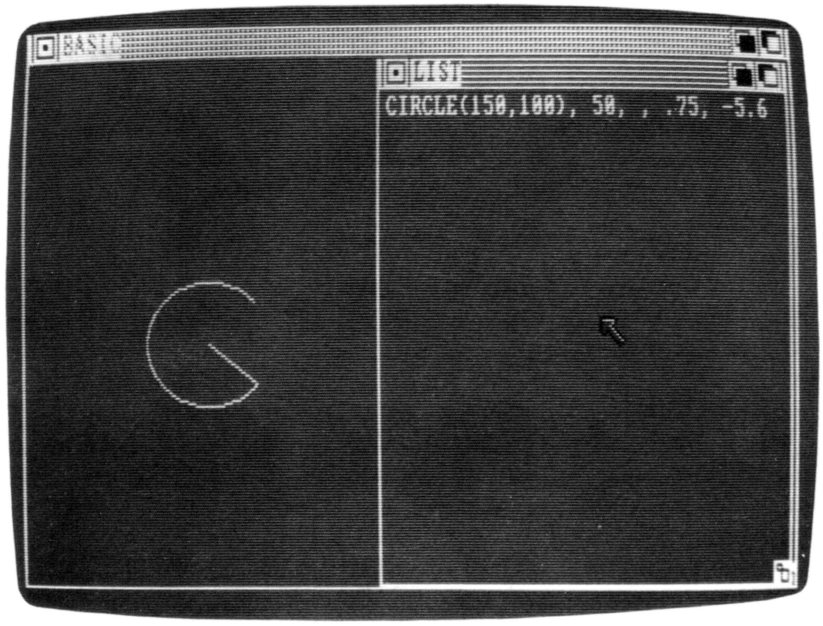


Figura 5-9.

Il comando CIRCLE può disegnare archi e aggiungere linee che congiungono il punto di partenza e/o di arrivo dell'arco al centro della circonferenza.

Per molti è assai più facile misurare la circonferenza in gradi piuttosto che in radianti. Per trasformare i gradi in radianti si può utilizzare la seguente formula:

```
(3.14159 * (gradi)) / 180 = radianti
```

Per esempio, un angolo di 45 gradi corrisponde a .785 (0,785) radianti.

Nella Figura 5-10 è riportato un cerchio misurato in radianti.

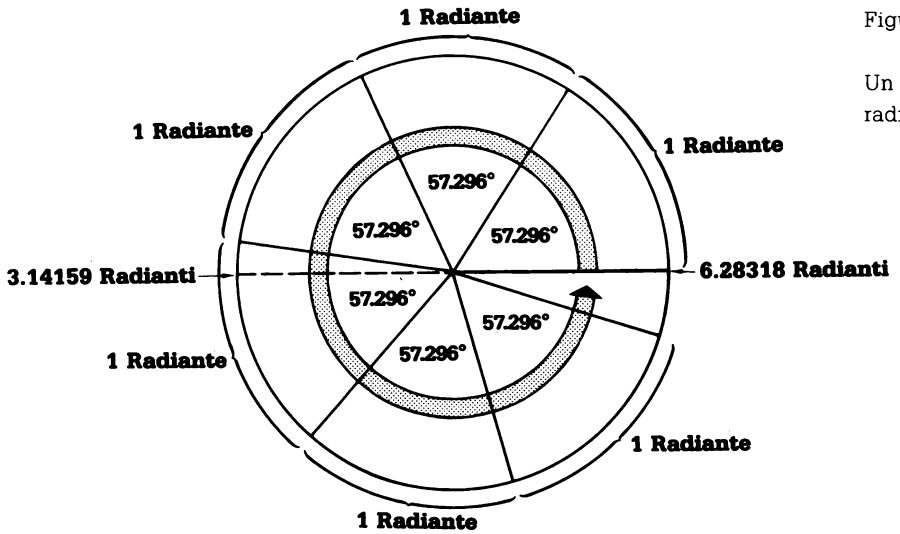


Figura 5-10.

Un cerchio misurato in radianti.

LA CREAZIONE DI ELLISSI

Modificando la sintassi del comando CIRCLE, è possibile variare l'altezza della circonferenza, e trasformarla in un'ellisse. Il parametro aspetto (che determina il cambiamento) è posto, nella sintassi del comando CIRCLE, subito dopo i valori che indicano gli estremi degli archi.

La formula usata da CIRCLE per creare una circonferenza parte dal presupposto che i pixel sullo schermo del monitor siano perfettamente quadrati; ma anche nei modi di risoluzione 1 e 4 i pixel sono leggermente più alti che larghi. Per ovviare a tale inconveniente, l'Amiga BASIC usa generalmente un valore prefissato per il parametro aspetto, il quale fa in modo che i cerchi appaiano perfettamente rotondi.

Se non viene specificato alcun parametro aspetto, CIRCLE utilizza automaticamente il valore .44 (0,44), che su uno schermo a risoluzione 2 disegna cerchi perfetti. Se si vuole modificare l'aspetto dei pixel, è sufficiente specificare un nuovo valore per l'aspetto. Se non vi interessa creare anche un arco, potete tralasciare di indicarne gli estremi, ma dovete comunque mettere delle virgole al loro posto. Ad esempio, questo comando aumenta il valore del parametro aspetto per allungare il cerchio tracciato in precedenza:

```
CIRCLE(150,100) , 50 , , , , 1
```

La Figura 5-11 ne illustra i risultati.

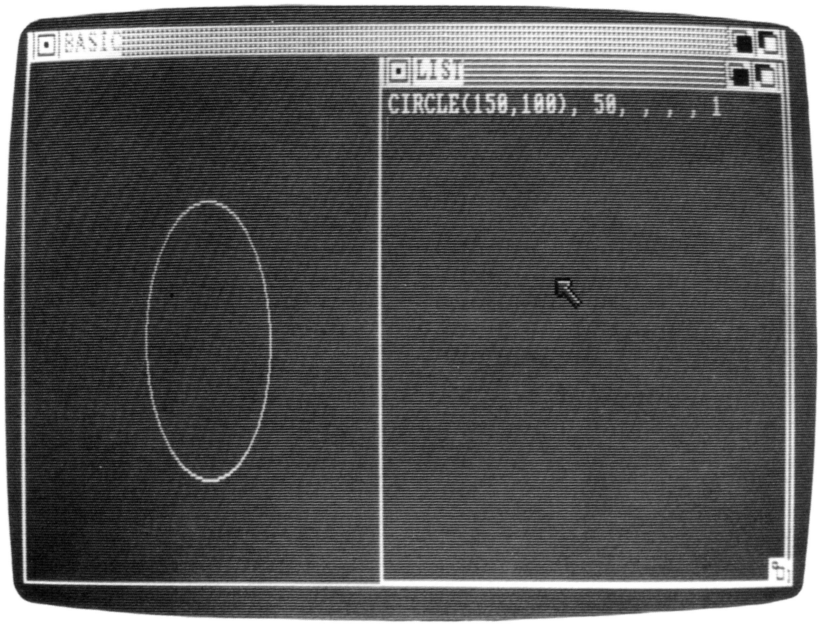


Figura 5-11.

Ellisse creata col comando CIRCLE.

Se si aumenta il valore dell'aspetto oltre il valore di default, la circonferenza si allunga in direzione verticale; se il nuovo aspetto è invece inferiore al valore di default, il cerchio si allunga orizzontalmente.

Dato che ogni modo di risoluzione utilizza pixel di diverse dimensioni, è necessario usare un aspetto diverso per ogni tipo di risoluzione, se si vogliono avere sempre cerchi perfetti. In uno schermo a risoluzione 2 verranno visualizzati cerchi automaticamente perfetti, poiché il valore di default dell'aspetto nell'Amiga BASIC è preselezionato per il secondo modo di risoluzione. Invece, gli schermi a risoluzione 1, 3 e 4 richiedono che l'utente specifichi il valore necessario a ottenere circonfe-

renze perfette. La seguente tabella riporta i valori del parametro aspetto adatti ad ogni singola risoluzione.

Risoluzione	Aspetto
1	.88
2	.44
3	1.76
4	.88

Se l'utente fissa il valore dell'aspetto e lo fa precedere anche dai valori necessari a creare un arco, il comando CIRCLE disegnerà archi che rispecchiano la curvatura dell'ellisse, come se si trattasse di un normalissimo cerchio.

LA CREAZIONE DI FIGURE POLIGONALI

AREA e AREAFILL sono due comandi BASIC usati simultaneamente per creare figure poligonali e colorarne l'interno con colori o matrici grafiche particolari. Questi comandi funzionano un po' come quei giochi in cui bisogna unire una serie di punti per far apparire un'immagine: si fissano col comando AREA i punti invisibili all'interno della finestra output, e poi si usa AREAFILL per collegare i punti e riempire l'area circoscritta con un colore o una matrice grafica.

IL COMANDO AREA

Il comando AREA ha la seguente sintassi:

```
AREA (indirizzo)
```

L'indirizzo può essere assoluto o relativo.

AREA è facile da usare. È sufficiente immettere un comando AREA per ogni punto che volete fissare all'interno della finestra. Per esempio, il comando

```
AREA(120,95)
```

fissa un punto invisibile all'indirizzo (120,95).

Per creare figure complesse è necessario usare una serie di comandi AREA che determinino i punti relativi ai lati della figura. Non si possono fissare più di 20 punti senza usare il comando AREAFILL che li colleghi. Nonostante questo, una volta collegati i punti col comando AREAFILL, è possibile

continuare a fissare nuovi punti per la stessa figura, oppure iniziarne una nuova.

Spesso è più semplice usare indirizzi relativi col comando AREA, naturalmente dopo aver fissato un primo punto con un indirizzo assoluto. Nell'esempio che segue vengono fissati i punti per creare un rombo. Il primo indirizzo è assoluto e i tre successivi sono relativi:

```
AREA(150,50)
AREA STEP(100,50)
AREA STEP(-100,50)
AREA STEP(-100,-50)
```

IL COMANDO AREAFILL

I comandi AREA di per sé non hanno alcun effetto visibile nella finestra output; si limitano a fissare i punti della figura che l'utente intende tracciare. È il comando AREAFILL che collega i punti con delle linee e che riempie la superficie interna della figura con il colore di primo piano di default, o con un colore o motivo scelto dall'utente. Se l'utente intende creare una figura vuota, deve tracciare ogni lato del perimetro singolarmente, ricorrendo ad altri comandi grafici. Questa è la sintassi di AREAFILL:

```
AREAFILL modo
```

Il modo è un'opzione che indica come riempire la superficie; può essere omesso qualora si desideri utilizzare il colore di default.

Per funzionare, AREAFILL deve essere preceduto da almeno due comandi AREA. AREAFILL collega dapprima i punti fissati con il comando AREA, nell'ordine in cui i punti sono stati immessi. Poi, raggiunto l'ultimo punto, lo collega automaticamente al primo, in modo che la serie di comandi AREA non debba terminare necessariamente con una ripetizione del primo comando.

Una volta collegati i punti, AREAFILL riempie la superficie interna e i margini col colore di default, se non si è specificato un modo diverso. Dato che AREAFILL colora tanto l'area quanto i margini della figura, questa risulta di un unico colore, e non può avere il perimetro delimitato da un colore diverso.

Per cambiare il colore della figura è necessario specificare il parametro modo. Se questo è uno 0, la figura sarà colorata col motivo creato dall'utente con il comando PATTERN (trattato in seguito), oppure, se non è stata creata alcuna matrice grafica, con il colore di primo piano di default. Immettendo come valore del parametro modo un 1, AREAFILL converte il colore di tutti i pixel racchiusi nell'area delimitata nel corrispondente colore inverso. Il corrispondente inverso di un colore è la tonalità del registro situato all'estremo opposto della palette a cui il colore in questione appartiene. La tabella riportata qui di seguito indica i colori inversi di uno schermo a otto colori (3 bit-plane).

Registro di colore	Registro di colore inverso
0	7
1	6
2	5
3	4
4	3
5	2
6	1
7	0

Se il parametro modo non viene specificato, AREAFILL presuppone che sia uno 0 e quindi usa il motivo creato con PATTERN (se quest'ultimo è assente, utilizza il colore di primo piano di default).

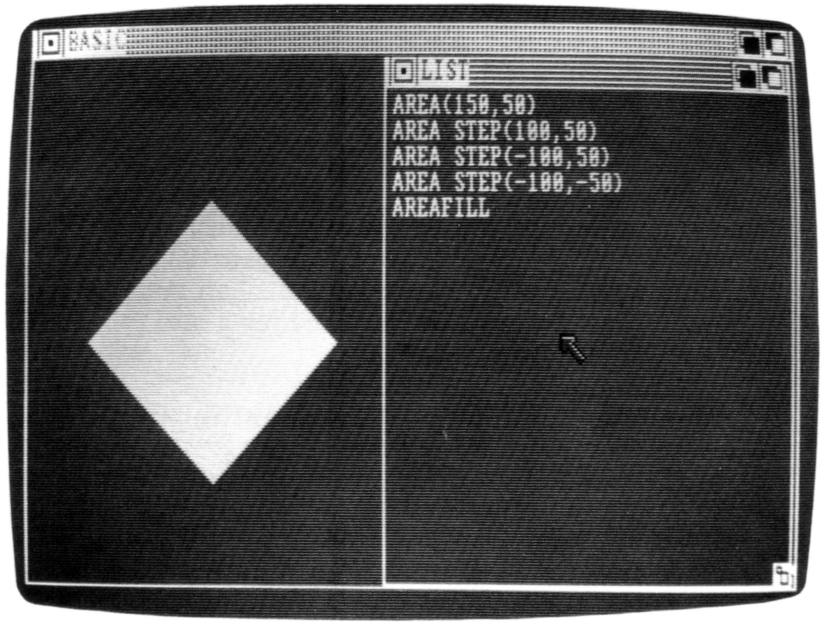
Per vedere come funziona AREAFILL, è sufficiente aggiungerlo ai quattro comandi AREA usati in precedenza per fissare i punti di un rombo e impartire quindi il run al programma:

```
AREA(150,50)
AREA STEP(100,50)
AREA STEP(-100,50)
AREA STEP(-100,-50)
AREAFILL
```

Sullo schermo apparirà il rombo colorato con la tonalità di default (vedere la Figura 5-12 riportata nella pagina successiva).

Figura 5-12.

Rombo creato coi comandi AREA e AREAFILL.



Se sostituite l'ultima linea del programma con

```
AREAFILL 1
```

e fate girare di nuovo il programma, vedrete apparire sullo schermo la stessa figura di prima, colorata però col colore inverso dei pixel dello sfondo. Non essendoci pixel di altri colori all'interno dell'area delimitata, AREAFILL può invertire solo il colore dei pixel dello sfondo.

Quando si creano figure con i comandi AREA e AREAFILL, niente impedisce di far sì che le linee del perimetro si incrocino fra di loro. Ad esempio, il programma seguente disegna due triangoli, col semplice espediente di far incrociare i due margini laterali. La Figura 5-13 ne illustra i risultati:

```
AREA(60,50)
AREA STEP(200,0)
AREA STEP(-200,100)
AREA STEP(200,0)
AREAFILL
```

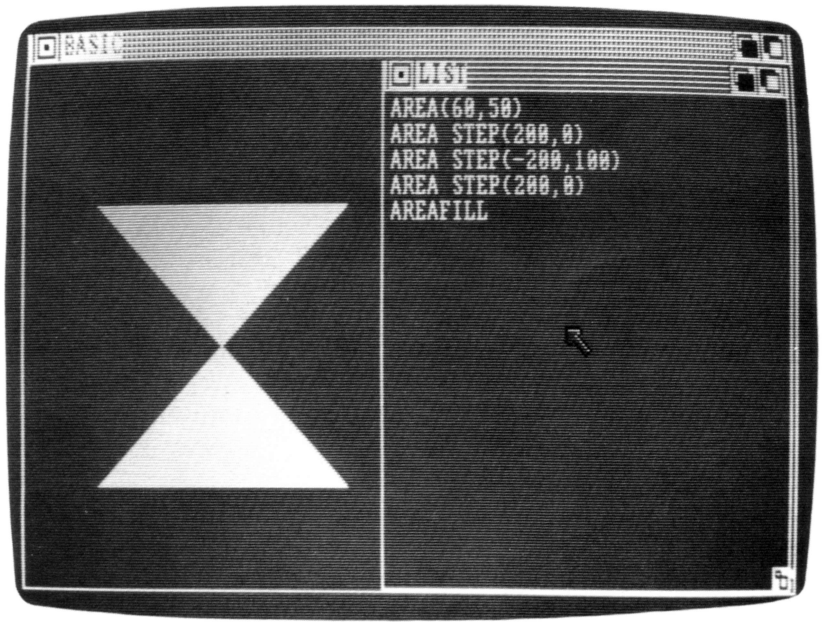


Figura 5-13.

I comandi AREA consentono di far incrociare i lati della figura.

COME RIEMPIRE LA SUPERFICIE INTERNA DI UNA FIGURA CON UNA MATRICE GRAFICA

Se l'utente ha realizzato una matrice grafica con il comando PATTERN (descritto più avanti in questo stesso capitolo), AREAFILL riempirà la figura disegnata servendosi di quel motivo, combinando il colore di primo piano con quello di sfondo, anziché limitarsi a usare solo il primo. Se come valore del parametro modo immettete 1, i colori di sfondo e primo piano saranno visualizzati dal BASIC tramite il ricorso al codice-colore 0 per lo sfondo e all'ultimo codice-colore (il colore inverso di 0) per il primo piano, con un bordo largo un pixel, lungo il perimetro della figura, del colore dello sfondo.

IL COMANDO PAINT

Abbiamo già visto come il comando LINE offra la possibilità di colorare l'area di figure vuote. Allo stesso modo è possibile usare il comando AREAFILL per colorare o decorare con matrici grafiche le figure tracciate con i comandi AREA. L'Amiga

BASIC permette altresì di riempire qualsiasi tipo di figura chiusa creata, ad esempio, con i comandi CIRCLE, PSET o LINE. Ciò è possibile grazie al comando PAINT. Ecco la sua sintassi:

PAINT *(indirizzo), colore-area, colore-bordo*

L'indirizzo può essere assoluto o relativo. I parametri colore-area e colore-bordo sono facoltativi; se inclusi, devono essere separati da virgole. Omettendo tali valori, PAINT colorerà l'area con il colore di primo piano di default. Se si desidera invece colorarla con un'altra tonalità, bisogna aggiungere una virgola dopo l'indirizzo e un codice-colore. PAINT riempierà la figura col colore del registro indicato.

Quando si usa il comando PAINT va anche tenuto presente il fatto che esso deve essere utilizzato in una finestra detta "refreshing window", cioè una finestra creata usando un valore per le caratteristiche delle finestre pari a 16 o più.

Prima di usare PAINT l'utente deve tracciare la sagoma della figura con un qualsiasi comando grafico (sia esso CIRCLE, LINE o PSET), assicurandosi inoltre che la linea tracciata sia continua e di un solo colore. In caso contrario PAINT, nel colorare la figura, uscirebbe dal bordo nei punti in cui questo è interrotto, oppure nei punti in cui la linea cambia colore, e invaderebbe l'intero schermo.

Nella sintassi del comando PAINT, l'indirizzo indica il punto in cui si desidera iniziare la colorazione. Tale punto può essere all'interno, ma anche all'esterno, della figura. PAINT colorerà lo schermo col colore di primo piano finché non incontrerà il perimetro della figura o i margini della finestra che la contiene. Iniziando da un punto all'interno dell'area della figura, PAINT riempirà l'area lasciando la superficie circostante del suo colore originale. Iniziando invece da un punto all'esterno del perimetro della figura, il comando PAINT colorerà tutto lo schermo, lasciando solo la figura del colore dello sfondo.

Il breve programma che segue disegna il cerchio riportato nella Figura 5-14 e lo riempie del colore di primo piano presente in memoria.

```
CIRCLE(140,90) , 100
PAINT(140,90)
```

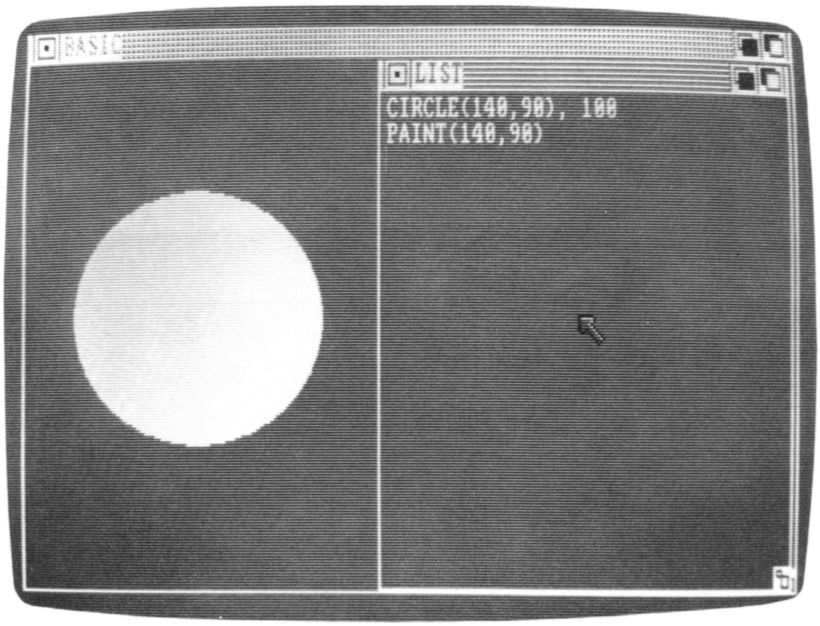


Figura 5-14.

Cerchio riempito con un comando PAINT.

Per riempire una figura con un colore diverso da quello di primo piano di default, è sufficiente aggiungere il codice-colore della tonalità desiderata. Il seguente comando PAINT fa uso del colore del registro 2:

```
PAINT(140,90) , 2
```

Usando questo comando PAINT nell'ultimo cerchio creato, otterrete il risultato di colorare l'intera superficie della finestra output. Il motivo è che PAINT riconosce i margini solo se sono dello stesso colore usato per l'area. Per riempire l'area con un colore diverso da quello del suo perimetro, è necessario mettere una virgola dopo il codice-colore e aggiungere il codice del colore prescelto per il margine. Per esempio, con un comando di questo genere, PAINT colorerà il cerchio con la tonalità del registro 3 riconoscendo però il margine colorato col registro 1:

```
PAINT(120,100), 3, 1
```

Immettendo e facendo girare il seguente programma, la circonferenza del cerchio della Figura 5-15 sarà disegnata col colore di primo piano, mentre l'area interna sarà del colore 3:

```
CIRCLE(120,100), 100
PAINT(120,100), 3, 1
```

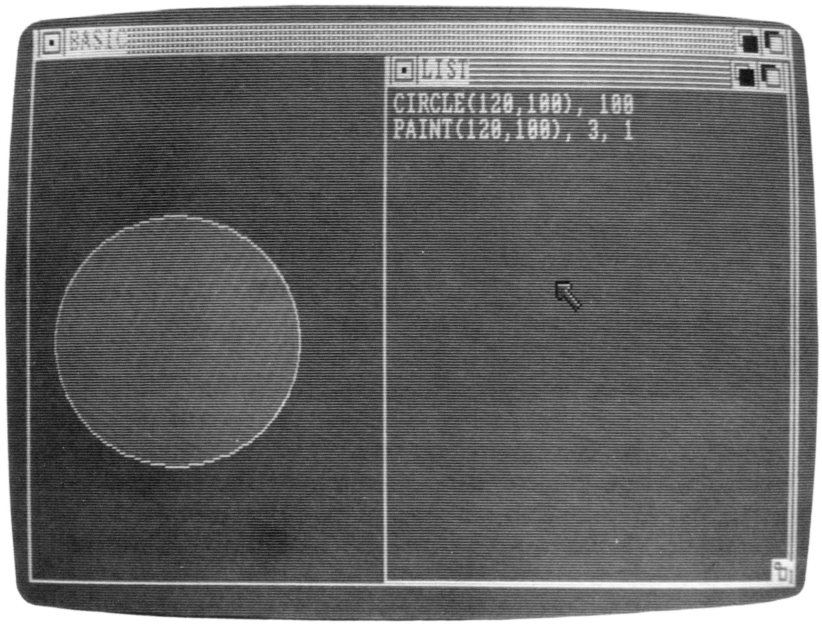


Figura 5-15.

Cerchio con la circonferenza nel colore di primo piano e l'area nel colore del registro numero 3.

Se l'utente utilizza il comando PATTERN (descritto nel prossimo paragrafo) per realizzare una matrice grafica, PAINT, invece di usare un unico colore, riempie la figura col motivo creato dall'utente, facendo uso del colore di default. Se viene specificato un codice-colore, il comando PAINT adopererà comunque il motivo disegnato col comando PATTERN, ma colorandolo con la tonalità indicata nel codice-colore specificato.

IL COMANDO PATTERN

Quando usate l'Amiga BASIC per tracciare linee e riempire aree, avete generalmente a che fare con linee continue e colori uniformi. Per variare e decorare con fantasia le vostre creazioni grafiche potete usare il comando PATTERN. Questo comando

crea disegni particolari che potete utilizzare per tracciare linee o riempire superfici. Una volta create e fissate queste matrici grafiche, qualsiasi comando grafico segua un comando PATTERN ne utilizzerà i motivi, sostituendoli alle solite righe continue e colorazioni uniformi.

Quella che segue è la sintassi del comando PATTERN:

PATTERN *maschera per le linee, nome array della maschera per le matrici grafiche*

È possibile omettere uno dei due parametri, ma non entrambi contemporaneamente.

CREAZIONE DI UNA MASCHERA PER LE LINEE

Per creare una linea-pattern è necessario creare prima una maschera a 16 bit, vale a dire un numero binario (in base 2) formato da 16 cifre, esclusivamente uno e zero. Ogni uno della maschera sta per il colore di primo piano e ogni zero per quello di sfondo. Il numero binario 1010101010101010, ad esempio, è una maschera che alterna i due colori ogni pixel. Stabilita la maschera per le linee, qualsiasi linea tracciata dall'Amiga BASIC continuerà a usarne il motivo, da sinistra verso destra. Ciò significa che se la linea tracciata è più lunga di 16 pixel, la maschera sarà utilizzata più di una volta, finché la linea è completamente disegnata. Se invece la linea misura meno di 16 pixel, l'Amiga BASIC, partendo dall'estremo sinistro, utilizzerà solo i pixel necessari.

Per creare una linea tratteggiata si può usare la maschera 1111000011110000, che alterna quattro pixel del colore di primo piano e quattro del colore dello sfondo. Se preferite una linea più continua con tratteggi più lunghi, provate con 1111111111110000. Si possono poi fare vari tentativi per trovare la combinazione più adatta alla maschera che volete realizzare col comando PATTERN.

Conversione di una maschera per le linee in un numero esadecimale

Il comando PATTERN non accetta parametri espressi in numeri binari, in quanto non è in grado di riconoscere numeri in base 2. L'Amiga BASIC interpreterebbe un numero binario di 16 bit come una cifra astronomica, nell'ordine dei quadrilioni. È dunque necessario trasformare la maschera binaria in un numero decimale (in base 10) o esadecimale (in base 16), affinché il comando PATTERN sia in grado di recepirla. È molto semplice convertire un un numero binario in un numero esadecimale che PATTERN sia in grado di riconoscere; basta

infatti dividere la maschera in gruppi di quattro bit e interpretare ciascun gruppo in base alla tabella riportata qui di seguito:

Binario	Esadecimale	Binario	Esadecimale
0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

Per esempio, la maschera 1110011111100111 viene suddivisa in 1110 0111 1110 0111. In base alla tabella i quattro gruppi equivalgono a E, 7, E e 7, per cui il numero esadecimale completo è E7E7. Inoltre, nell'Amiga BASIC, i numeri esadecimali sono sempre preceduti da &H, quindi la maschera, espressa in esadecimali, diventa &HE7E7. Se &H non viene specificato, il comando PATTERN presume che l'utente stia usando un numero decimale (in base 10). Pur avendo la possibilità di usare i decimali, vi accorgerete che è più semplice lavorare con valori esadecimali.

Come applicare una maschera per le linee a un comando PATTERN

Una volta che si è ideata una matrice lineare, e la si è trasformata in una maschera binaria e poi in numero esadecimale, è possibile attivarla con l'ausilio di un comando PATTERN. Il breve programma che segue disegna un riquadro facendo uso della maschera creata in precedenza:

```
PATTERN &HE7E7
LINE(65,40) - STEP(150,100), , B
```

Nella Figura 5-16 è riportato il riquadro tracciato dall'Amiga BASIC con il tipo di linea definito dalla maschera &HE7E7. Se continuate ad usare il comando LINE, l'Amiga BASIC continuerà a usare lo stesso motivo per tracciare qualsiasi altra linea o figura di vostra creazione; non subiscono invece alcuna modifica i cerchi e gli archi disegnati col comando CIRCLE.

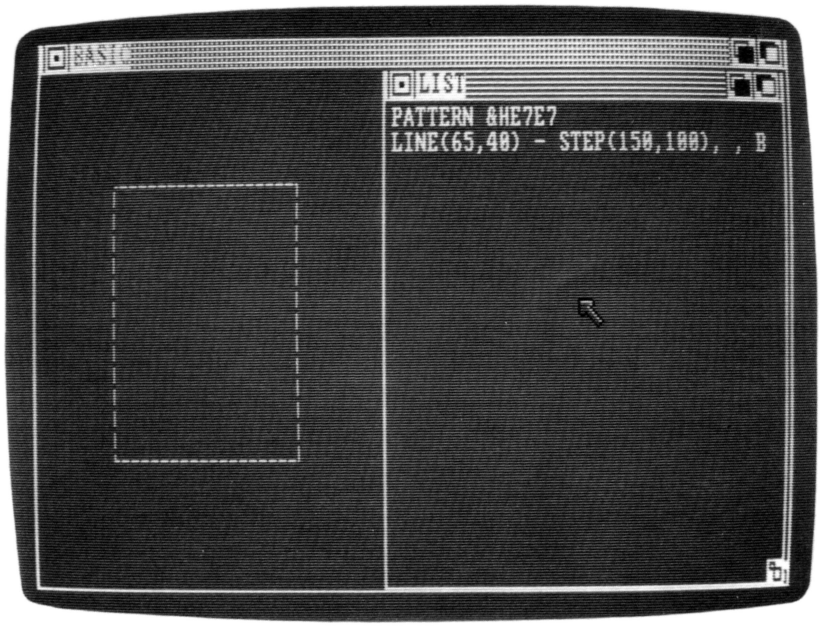


Figura 5-16.

Riquadro col perimetro tratteggiato creato grazie al comando PATTERN.

Per riottenere una linea continua, utilizzate:

```
PATTERN &HFFFF
```

che definisce una maschera a linea continua (tutti uno).

CREAZIONE DI UN'AREA PATTERN

Una maschera lineare ha una sola dimensione ed è possibile realizzarla creando un'unica maschera binaria di 16 bit, ovvero una stringa di uno e zero. Per creare un'area pattern bisogna definire una maschera bidimensionale usando un array di uno e zero. Per creare un'area, PATTERN fa uso di un array sempre largo 16 bit e alto 2 o una potenza di due. Per creare la maschera è utile abbozzarla prima su un foglio di carta millimetrata divisa in colonne larghe 16 quadratini e alte a piacere (purché l'altezza sia sempre una potenza di due). Dopodiché segnate 1 dove volete che appaia il colore di primo piano, e 0 dove volete il colore di sfondo. Alla fine dovrebbe risultare una colonna di numeri binari da 16 bit.

La maschera bidimensionale qui riportata crea ad esempio un motivo a pois quadrati:

```
0000111100001111
0000111100001111
0000000000000000
0000000000000000
1111000011110000
1111000011110000
0000000000000000
0000000000000000
```

Come trasformare la maschera di un'area pattern in un array intero

Affinché il comando PATTERN recepisca la maschera bidimensionale, è necessario trasformarla in un array intero. Per farlo si deve innanzitutto tradurre ogni linea in un numero esadecimale, proprio come nel caso precedente della maschera lineare. La maschera dell'area a pois appena definita si traduce così:

```
0F0F
0F0F
0000
0000
F0F0
F0F0
0000
0000
```

Dopo aver tradotto ogni linea della maschera, si conta il numero delle linee e si introduce nel programma un array intero che utilizzi tanti elementi quante sono le linee. Per il motivo a pois del nostro esempio è necessario specificare otto elementi (uno per ogni linea). Quindi bisogna dimensionare un array a 7 con un comando DIM. Dato che ogni array inizia con l'elemento numero 0, di fatto si hanno 8 elementi in un array dimensionato a 7. L'utente deve poi assegnare le linee della maschera agli elementi dell'array, partendo dall'alto. Le linee del programma riportato nella pagina successiva assegnano la maschera a pois a un array *POLKADOT%* (laddove il % indica l'array).

```
DIM POLKADOT%(7)
POLKADOT%(0) = &H0F0F
POLKADOT%(1) = &H0F0F
POLKADOT%(2) = &H0000
POLKADOT%(3) = &H0000
POLKADOT%(4) = &HF0F0
POLKADOT%(5) = &HF0F0
POLKADOT%(6) = &H0000
POLKADOT%(7) = &H0000
```

Come applicare un'area pattern in un comando PATTERN

Una volta creata, l'area pattern viene applicata semplicemente aggiungendo il nome dell'array dopo il parametro pattern-linea del comando PATTERN, e separando i due valori con una virgola. Se non intendete specificare il primo parametro, sostituitelo con una virgola e poi inserite il nome dell'array. Qualsiasi comando grafico dell'Amiga BASIC venga usato d'ora in poi per riempire una superficie farà uso del motivo grafico da voi indicato. Il seguente comando mette in funzione la maschera appena ideata:

```
PATTERN, POLKADOT%
```

Il seguente programma riunisce il tutto per disegnare un riquadro interamente decorato da pois quadrati (polka dot), come illustrato nella Figura 5-17 (nella pagina successiva):

```
DIM POLKADOT%(7)
POLKADOT%(0) = &H0F0F
POLKADOT%(1) = &H0F0F
POLKADOT%(2) = &H0000
POLKADOT%(3) = &H0000
POLKADOT%(4) = &HF0F0
POLKADOT%(5) = &HF0F0
POLKADOT%(6) = &H0000
POLKADOT%(7) = &H0000
PATTERN, POLKADOT%
LINE(70,10) - STEP(150,160), , BF
```

Figura 5-17.

Riquadro a pois quadrati creato coi comandi LINE e PATTERN.

Digitando voi stessi il programma, vi accorgerete che il BASIC abbrevia i numeri immessi per ogni elemento dell'array. Per esempio, se scrivete &H0000 e premete RETURN, il BASIC trasforma il numero in &H0, che ha lo stesso valore.

Per restituire all'area pattern un colore uniforme, è sufficiente creare un semplice array da due elementi, assegnare una maschera a motivo uniforme e poi inserirla in un comando PATTERN:

```
DIM SOLID%(1)
SOLID%(0) = &HFFFF
SOLID%(1) = &HFFFF
PATTERN, SOLID%
```

COME CAMBIARE I COLORI DI UNA MATRICE GRAFICA

Ricordate che non siete limitati nella scelta dei colori quando usate matrici grafiche per le linee e per le aree (pattern line ed area). Gli 1 e gli 0 della maschera indicano solo quali bit saranno del colore di primo piano e quali del colore di sfondo. Si può sempre ricorrere al comando COLOR per variare i colori indicando registri diversi, oppure si può utilizzare il comando PALETTE per cambiare i colori contenuti nei registri dello sfondo e del primo piano.

Questo capitolo ha presentato i comandi che colorano i pixel, tracciano linee e creano figure, riempiendole anche di matrici grafiche. Combinando questi comandi si possono creare immagini complesse sullo schermo. Nel prossimo capitolo vedremo com'è possibile aggiungere testo alle immagini, inserire le immagini in finestre di dimensioni variabili e copiare parte di una figura per trasferirla in una posizione diversa della finestra.

SESTO CAPITOLO
LA GRAFICA E L'AMIGA BASIC: MISCELLANEA

L'Amiga BASIC è dotato di un'ampia serie di comandi e funzioni grafiche che svolgono compiti diversi dal creare schermi e finestre per poi riempirle di elementi grafici. Questa miscellanea di comandi e funzioni consente di classificare le vostre creazioni, modificarne le misure a seconda delle dimensioni della finestra utilizzata, oppure trasferirle da un settore dello schermo a un altro.

Molti di questi comandi e funzioni possono essere utilizzati per lavorare sui testi in modo creativo; è possibile aggiungere testi alle immagini grazie al comando PRINT, e, con i comandi LOCATE e COLOR, posizionare il testo nel modo voluto e con i colori preferiti. Per tenere continuamente sotto controllo la locazione del testo sono state predisposte le funzioni CSRLIN e POS(0).

Altri comandi e funzioni aiutano l'utente ad adattare le creazioni grafiche alle dimensioni mutevoli di una finestra ridefinibile. All'interno di un programma, la funzione WINDOW() consente di conoscere l'altezza e la larghezza della finestra, informazioni che si possono utilizzare per adattare le immagini alle nuove dimensioni della finestra quando questa viene modificata dall'utente attraverso il gadget di dimensionamento. La funzione WINDOW() serve anche a controllare il numero di ID della finestra, a vedere quanti colori possono essere utilizzati ed esaminare altre informazioni che riguardano la finestra stessa. Per conoscere il colore di un singolo pixel all'interno di una finestra si può usare la funzione POINT.

I comandi GET e PUT offrono la possibilità di manipolare gli elementi grafici o loro parti all'interno di una finestra. Con GET e PUT è inoltre possibile far apparire in una finestra output una serie di immagini in rapida successione.

INDIRIZZAMENTO DEI CARATTERI

Quando si introducono parti di testo in una finestra output (come commento alle immagini), è fondamentale conoscere la procedura necessaria a posizionare i caratteri con cura. Per questo è indispensabile comprendere a fondo il modo in cui bisogna usare l'indirizzamento dei caratteri.

Per determinare la posizione dei caratteri di un testo, si usa un sistema di indirizzamento completamente diverso da quello visto per l'indirizzamento dei pixel. Invece di contare linee e colonne di pixel, si contano linee e colonne di testo. Invece di

muovere un invisibile cursore grafico da un pixel all'altro, si muove un invisibile cursore testo da un carattere all'altro.

Gli indirizzi dei caratteri iniziano con una coppia di numeri diversi da quelli usati per i pixel; le linee e le colonne, invece di partire da 0, partono dal numero 1. Ad esempio, il carattere situato nell'angolo superiore sinistro corrisponde all'indirizzo 1,1, non alle coordinate (0,0). La numerazione degli indirizzi per i testi inizia nell'angolo superiore sinistro della finestra, e il numero cresce a mano a mano che ci si sposta verso destra o verso il basso, allo stesso modo del sistema di numerazione degli indirizzi dei pixel. La Figura 6-1 illustra il modo in cui sono numerate le linee e le colonne.

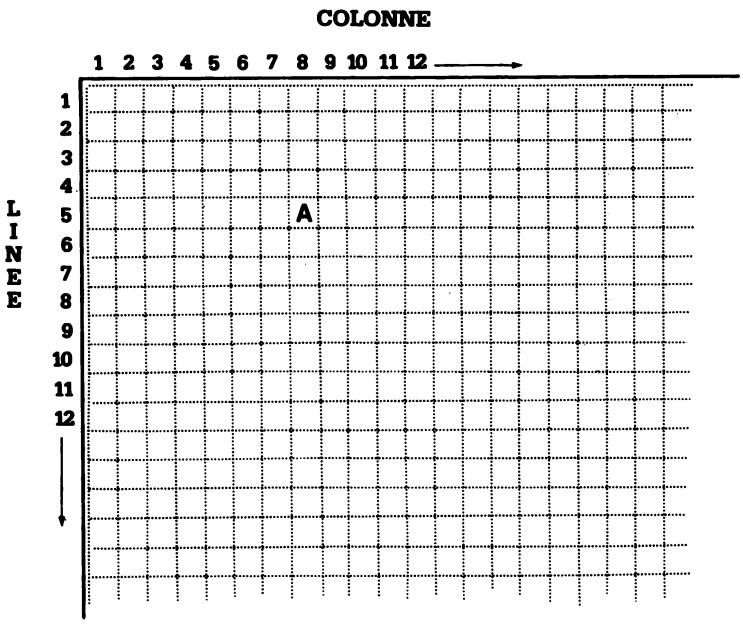


Figura 6-1.

Numerazione in linee e colonne per l'indirizzamento dei caratteri.

Invece di specificare prima la colonna e poi la linea come nel caso dei pixel, un indirizzo di caratteri specifica prima la linea e poi la colonna. I due numeri sono separati da una virgola ma non devono essere racchiusi fra parentesi. Ad esempio, un carattere posizionato nella quarta linea a partire dall'alto e nella settima colonna da sinistra si trova all'indirizzo 4,7.

USO DI CORPI CARATTERE DIFFERENTI

Le dimensioni dei caratteri stabilite dalla fonte utilizzata con l'Amiga BASIC determinano il numero di linee e colonne di testo contenute nello schermo; il tutto, logicamente, va a

interessare i numeri usati nell'indirizzamento. Appena avviato, l'Amiga BASIC utilizza lo stesso corpo carattere che l'utente sta usando con il Workbench. Il programma Preferences presente nel dischetto del Workbench consente però di modificare la fonte che si vuole utilizzare sullo schermo.

L'utente ha due opzioni a sua disposizione: quella di default a 60 colonne e quella a 80 colonne. La prima utilizza caratteri che misurano 10 pixel in larghezza e 9 in altezza, dimensioni sufficienti a una visione nitida sia su apparecchio televisivo, sia su monitor composito. La fonte a 80 colonne utilizza un set di caratteri più piccoli per consentire l'inserimento di testi più lunghi su ogni singola riga. Questi caratteri misurano 8 pixel in altezza e 8 in larghezza, sono più stretti e più corti degli altri. È però opportuno sottolineare che i testi a 80 colonne non risultano chiaramente leggibili sui televisori o sui monitor compositi, mentre lo sono sui monitor RGB.

Le dimensioni dei caratteri contenuti in una finestra dipendono anche dalla risoluzione dello schermo in cui la finestra è inserita. Tutti i caratteri di una fonte hanno le stesse misure in pixel, ma dal momento che la grandezza dei pixel varia a seconda della risoluzione adottata, questo influisce anche nella grandezza dei caratteri. Ad esempio, i caratteri stampati su uno schermo a bassa risoluzione sono larghi il doppio di quelli su schermi ad alta risoluzione. I caratteri di uno schermo a pixel non-interlace sono alti il doppio rispetto a quelli riprodotti su uno schermo a pixel interlace.

La combinazione di fonte e risoluzione determina il numero massimo di caratteri contenuti in una finestra che occupa l'intera superficie dello schermo. Lo schema seguente indica il numero di linee e colonne di una finestra a tutto schermo con una linea d'intestazione e un gadget di dimensionamento.

	Modo 1	Modo 2	Modo 3	Modo 4
Fonte 60 colonne:	20 lin.	20 lin.	42 lin.	42 lin.
	30 col.	61 col.	30 col.	61 col.
Fonte 80 colonne:	23 lin.	23 lin.	48 lin.	48 lin.
	38 col.	77 col.	38 col.	77 col.

Nella Figura 6-2 è riportato un monitor sul quale appaiono contemporaneamente schermi a quattro risoluzioni diverse. Ogni schermo visualizza la stessa frase utilizzando la stessa fonte. Eppure, notate la diversità delle dimensioni dei caratteri.

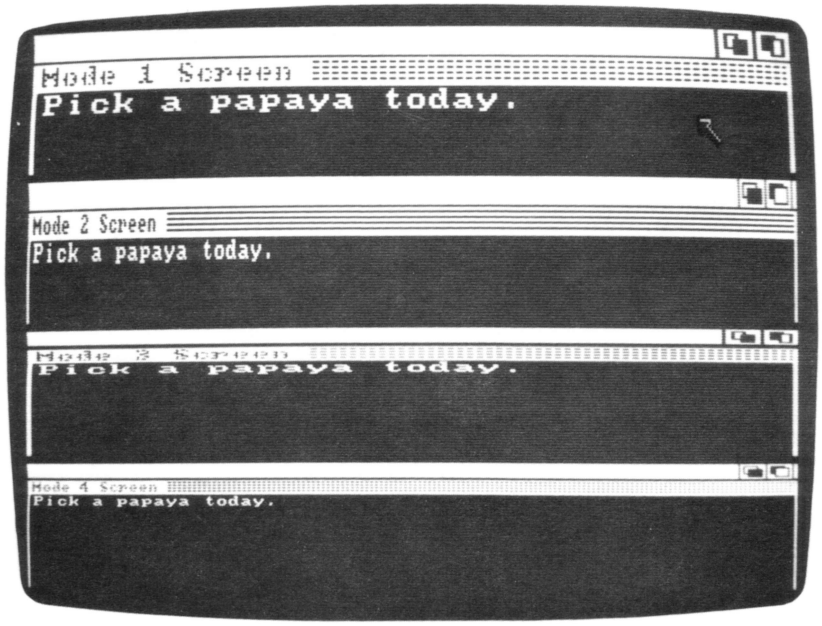


Figura 6-2.

Le dimensioni dei caratteri del testo dipendono dalla risoluzione dello schermo su cui vengono visualizzati.

IL COMANDO PRINT

Chiunque conosca il BASIC avrà sentito parlare del comando PRINT, utilizzato per stampare messaggi sullo schermo. La sintassi del comando PRINT è:

PRINT *lista messaggi*

La lista può essere una qualsiasi combinazione di stringhe, costanti, variabili numeriche e variabili stringa, separate da una virgola o da un punto e virgola.

È importante sapere sempre dove il comando PRINT colloca il cursore dopo aver eseguito la sua funzione. Se la lista termina con una virgola, il cursore si muove lungo la stessa linea finché incontra la successiva virgola di stop, simile allo stop di tabulazione della macchina da scrivere. Se invece la lista dei messaggi termina con un punto e virgola, il cursore resta sempre sulla stessa linea ma si ferma alla fine del messaggio appena stampato. Se non si mette né la virgola né il punto e virgola, il cursore si sposta all'inizio della riga successiva. Informazioni più dettagliate sulla collocazione del cursore testo vengono fornite dal manuale dell'Amiga BASIC, nella sezione dedicata al comando PRINT.

IL COMANDO LOCATE

Quando si vogliono combinare testo e immagini in una sola finestra e si ha la necessità di stampare il testo a un indirizo ben preciso al fine di evitare sovrapposizioni, può risultare alquanto noioso (e a volte impossibile) collocare il cursore testo con numerosissimi comandi PRINT e rispettive virgole e punti e virgola. Non solo è necessario usare un comando PRINT per ogni linea stampata, ma se si scende troppo in basso, si rischia di far scattare uno scroll che metterebbe tutto sottosopra. Per posizionare il testo con precisione è preferibile utilizzare il comando LOCATE.

Il comando LOCATE specifica il punto dello schermo in cui il comando PRINT successivo dovrà iniziare a stampare. La sintassi è:

LOCATE *numero linea, numero colonna*

I due numeri di linea e colonna che specificano l'indirizzo sono opzionali, ma se non vengono inseriti, il comando LOCATE non riposiziona il cursore. Si possono specificare entrambi i numeri e ometterne uno o l'altro. Se non viene precisato il numero della linea, è comunque necessario mettere una virgola prima del numero della colonna. Se si specifica solo il numero della linea, la virgola non serve.

I comandi LOCATE riportati qui di seguito illustrano esempi dei tre possibili tipi di indirizzo:

Comando	Risultato
LOCATE 9, 24	Muove il cursore alla linea 9, colonna 24.
LOCATE 4	Muove il cursore alla linea 4 e utilizza il numero di colonna dell'indirizzo corrente del cursore.
LOCATE ,45	Muove il cursore alla colonna 45 e utilizza il numero di linea dell'indirizzo corrente del cursore.

ESEMPI DI COMANDO LOCATE

Usando il comando LOCATE con il comando PRINT è possibile posizionare accuratamente le sezioni di testo sullo schermo. Nel breve programma che segue, un ciclo FOR...NEXT cambia l'indirizzo del comando LOCATE per tutti i valori compresi fra

1,1 e 1,19, stampando una linea diagonale di parole che occupa tutto lo schermo:

```
FOR i = 1 TO 19
    LOCATE i, i
    PRINT ''Diagonal''
NEXT i
```

Il risultato è illustrato dalla Figura 6-3.

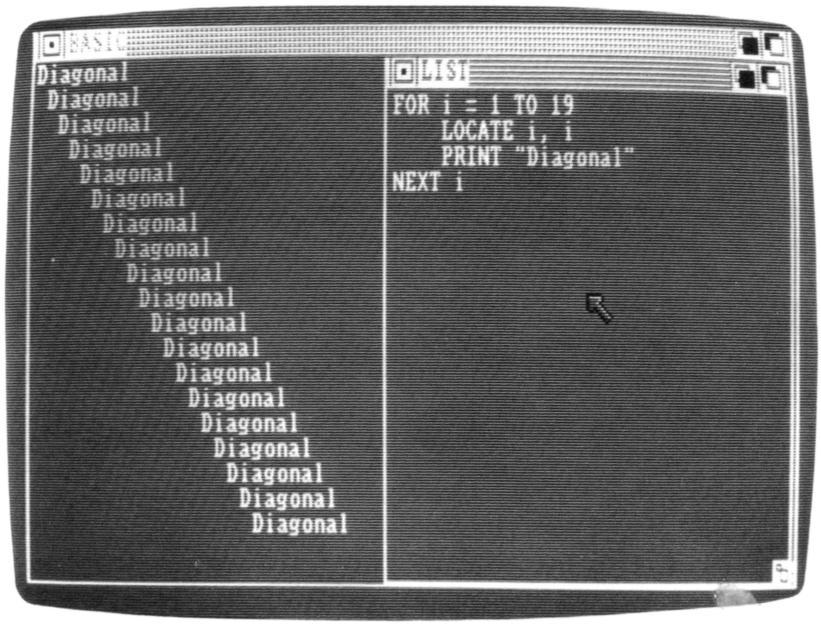


Figura 6-3.

Numerazione in linee e colonne per l'indirizzamento dei caratteri.

Un altro uso intelligente di LOCATE consiste nel mantenere il cursore testo sempre nella stessa posizione, cosicché il comando PRINT che lo segue stampi il nuovo messaggio sovrapponendolo a quello precedente. Il programma seguente visualizza i numeri da 1 a 100, stampandoli sempre sullo stesso punto dello schermo.

```
LOCATE 11,5
PRINT ''Siamo al numero:'';
FOR i = 1 TO 100
    LOCATE 11, 21
    PRINT i
NEXT i
```

POSIZIONAMENTO DEL CURSORE TESTO CON LE FUNZIONI CSRLIN E POS(0)

Può capitare che si abbia la necessità di allineare un nuovo testo con uno già visualizzato sullo schermo. In questo caso ci si può servire del comando LOCATE, ma può risultare difficile rintracciare l'esatta locazione. Ad esempio, per calcolare l'indirizzo del secondo LOCATE nell'ultimo programma, bisogna contare il numero dei caratteri della stringa *Siamo al numero:* e aggiungere tale cifra al numero del primo comando LOCATE, calcolando anche uno spazio dopo i due punti. È però possibile usare due funzioni, CSRLIN e POS(0), che fissano e memorizzano in qualsiasi momento l'indirizzo del cursore per utilizzarlo in un successivo comando LOCATE.

CSRLIN memorizza il numero di linea della posizione del cursore. Tale valore può essere immagazzinato come variabile numerica. Ad esempio,

```
x = CSRLIN
```

memorizza il numero di linea dell'attuale posizione del cursore sotto il nome della variabile *x*.

POS(0) memorizza il numero di colonna della posizione del cursore. Come la funzione precedente, il valore riportato da POS(0) può essere assegnato a una variabile. Ad esempio,

```
y = POS(0)
```

memorizza sotto il nome di variabile *y* il numero di colonna della locazione del cursore.

Usando le due funzioni contemporaneamente, è possibile memorizzare l'indirizzo del cursore testo in qualsiasi momento, utilizzando solo due variabili. Ad esempio, il programma che segue è una versione modificata del precedente. Il risultato è lo stesso, ma il secondo comando LOCATE non usa un indirizzo specifico; si serve della locazione del cursore testo rilevata alla fine del comando PRINT e immagazzinata attraverso le variabili *x* e *y*; il nuovo indirizzo fissato in questo modo è situato una colonna dopo la fine della stringa *Siamo al numero:* .

```
LOCATE 11, 5
PRINT ''Siamo al numero:'';
x = CSRLIN
y = POS(0)
FOR i = 1 TO 100
    LOCATE x, y
    PRINT i
NEXT i
```

Notate il punto e virgola alla fine del primo comando PRINT; serve a fermare il cursore alla fine della stringa appena stampata, in modo che CSRLIN e POS(0) possano leggerne l'indirizzo. Fate attenzione anche al secondo comando LOCATE, che posiziona il cursore subito dopo la fine della stringa *Siamo al numero:*. Il numero viene stampato con uno spazio di separazione. Ciò avviene perché il comando PRINT lascia sempre uno spazio bianco prima di un numero positivo. Per posizionare esattamente il testo, è importante conoscere sempre il punto in cui il comando PRINT ha lasciato il cursore.

COME USARE IL COMANDO COLOR CON PRINT

Nel capitolo precedente abbiamo visto come usare il comando COLOR per selezionare i colori di sfondo e di primo piano per i comandi grafici. Il comando COLOR può essere usato anche per selezionare i colori di un testo. Il comando PRINT, quando stampa dei caratteri sullo schermo, utilizza il colore di primo piano per i caratteri e il colore di sfondo per lo schermo sottostante.

Se immettete i seguenti comandi, sullo schermo apparirà il messaggio in caratteri di colore 3 su uno sfondo di colore 2:

```
COLOR 3,2
PRINT ''Colori nuovi!''
```

Se volete tornare ai colori di default scrivete:

```
COLOR 1, 0
```

Nel caso desideriate sovrapporre una scritta a una figura, è possibile selezionare per il testo un colore di sfondo identico alla tinta dell'immagine, fondendo in pratica caratteri e disegno; oppure potete selezionare un colore di sfondo contrastante per far risaltare la striscia di testo. Il programma listato qui di seguito disegna nella finestra output tre figure di colori diversi e utilizza poi il comando LOCATE per stampare delle targhette su ogni figura. I colori di sfondo e primo piano sono diversi per ogni targhetta; in due casi lo sfondo del testo ha lo stesso colore delle figure, mentre nel terzo esempio è stato selezionato un colore di sfondo che faccia risaltare il testo. I comandi LOCATE di questo programma usano indirizzi adatti allo schermo a 60 colonne. Se fate girare il programma con una fonte a 80 colonne, le targhette non saranno visualizzate al posto giusto.

```
SCREEN 1, 640, 200, 3, 2
WINDOW 2, ''Shapes '', , 15, 1

MakeCircle:
   CIRCLE(160,50), 100, 5
   PAINT(160,50), 5, 5

MakeTriangle:
   COLOR 6
   AREA(480,5)
   AREA STEP(120,90)
   AREA STEP(-240,0)
   AREAFILL

MakeRectangle:
   LINE(20,100) - (600,180), 4, BF

AddLabels:
   LOCATE 6,14
   COLOR 2, 5
   PRINT ''Circle''
   LOCATE 8, 45
   COLOR  2, 6
   PRINT ''Triangle''
   LOCATE 17, 26
   COLOR 2, 0
   PRINT ''Rectangle''
```

I risultati sono riportati nella Figura 6-4.

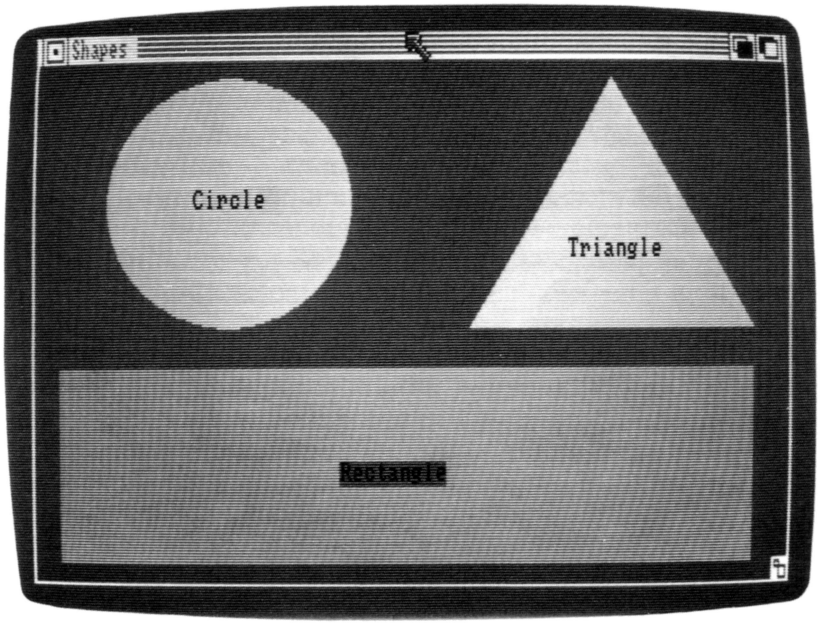


Figura 6-4.

Alle figure visualizzate è stato aggiunto testo utilizzando diversi colori di sfondo e primo piano.

INTRODUZIONE ALLA GRAFICA INTERATTIVA

La grafica degli esempi proposti finora presuppone un'area di lavoro stabile. La finestra deve essere sempre della stessa misura, il numero di bit-plane deve restare invariato e la risoluzione dello schermo deve limitarsi a un unico modo. Tuttavia non si può essere sempre limitati da un'area di lavoro stabile. Intuition consente all'utente di muovere le finestre sulla superficie dello schermo, cambiarne le dimensioni e variare altre caratteristiche dello schermo. Modificata l'area operativa, i vostri programmi dovranno creare una grafica interattiva, cioè una grafica che muta adattandosi di volta in volta alle modifiche apportate all'area che la contiene.

Per leggere le condizioni grafiche, l'Amiga BASIC è dotato di due funzioni, WINDOW( ) e POINT. È possibile usare i risultati di queste funzioni per creare un programma la cui grafica sia in grado di adattarsi a condizioni mutevoli; ad esempio, la dimensione delle figure può variare per adattarsi a

eventuali modifiche delle dimensioni della finestra, oppure il programma può stabilire quale fra numerose finestre è stata selezionata col mouse e iniziare quindi a lavorare in quella finestra. Potete anche usare le due funzioni WINDOW() e POINT per creare una subroutine che funzioni con risoluzioni e finestre di vari tipi. La subroutine verrà immagazzinata per essere usata in diversi programmi che ne possano sfruttare la grande versatilità.

Le funzioni possono essere usate anche per semplificare i calcoli degli indirizzi dei pixel; potendo leggere le misure di una finestra, l'utente sa subito dove sono collocati i margini e può quindi evitare di immettere un indirizzo situato all'esterno dei margini stessi. La lettura delle dimensioni della finestra facilita anche i calcoli necessari per stabilire quanti pixel guadagnerà o perderà l'area interna della finestra nel caso venga aggiunto un gadget o una linea d'intestazione.

LA FUNZIONE WINDOW()

Fate attenzione a non confondere la funzione WINDOW() col comando WINDOW. Il comando crea una finestra sullo schermo. La funzione legge invece nove diverse condizioni e le trasmette al programma. La funzione WINDOW() ha anche una sintassi completamente diversa:

```
WINDOW(numero condizione)
```

Il numero condizione può essere un numero intero positivo qualsiasi compreso fra 0 e 8, il quale determina la condizione che WINDOW() andrà a leggere. La funzione WINDOW() non produce alcun effetto se inserita in una linea priva di altre istruzioni; essa fornisce un valore che deve essere assegnato a una variabile o utilizzato in un comando.

La tabella alla pagina successiva riporta i nove numeri che determinano la condizione, con i rispettivi valori assegnati.

Numero condizione	Valore ritornato
0	Il numero di ID della finestra selezionata col mouse. La finestra selezionata è quella con l'intestazione illuminata, che spicca fra quelle delle altre finestre.
1	Il numero di ID della finestra output in cui PRINT e gli altri comandi grafici stanno lavorando.
2	La larghezza in pixel dell'area interna della finestra output utilizzata dall'utente.
3	L'altezza in pixel dell'area interna della finestra output utilizzata dall'utente.
4	Il numero di colonna dell'indirizzo del pixel nella finestra output situato nell'angolo inferiore sinistro del cursore testo. Unito al valore della condizione 5, consente di posizionare con precisione un testo nella finestra.
5	Il numero di linea dell'indirizzo del pixel nella finestra output situato nell'angolo inferiore sinistro del cursore testo.
6	Il codice-colore più elevato disponibile nella finestra output. Indica quanti colori potete utilizzare nella finestra.
7	Il puntatore alla finestra Intuition in funzione. Questo puntatore è un indirizzo di memoria usato dai programmatori nelle subroutine in linguaggio assembly.
8	Il puntatore alla Intuition Rastport corrente. Questo puntatore è un indirizzo di memoria usato dai programmatori nelle subroutine in linguaggio assembly.

ESEMPI DELLA FUNZIONE WINDOW()

Dato che WINDOW() è una funzione che, presa di per sé, non fa molto, il modo migliore per imparare a usarla con dimestichezza è quello di introdurla in alcuni programmi. Il programma riportato nella pagina successiva crea tre diverse finestre, contrassegnate dai numeri 2, 3 e 4, e seleziona la finestra numero 4 come finestra output, poiché è stata l'ultima a essere aperta. Il programma prosegue visualizzando il numero della finestra output in funzione e di quella selezionata, utilizzando WINDOW(1) e WINDOW(0) per leggere i valori corrispondenti. Se muovete il puntatore sullo schermo mentre il programma sta girando e selezionate diverse finestre, noterete il cambiamento dell'indicazione relativa al numero della finestra selezionata.

```
WINDOW 2, ''Window 2 '', (0,0) - (280,80), 15
WINDOW 3, ''Window 3 '', (320,0) - (600,80), 15
WINDOW 4, ''Window 4 '', (0,100) - (280,180), 15
PRINT ''Output window is'';
x1 = POS(0): y1 = CSRLIN
PRINT
PRINT ''Selected window is'';
x2 = POS(0): y2 = CSRLIN

Loop:
   LOCATE y1, x1
   PRINT WINDOW(1)
   LOCATE y2, x2
   PRINT WINDOW(0)
   GOTO Loop
```

Il programma crea uno schermo come quello della Figura 6-5.

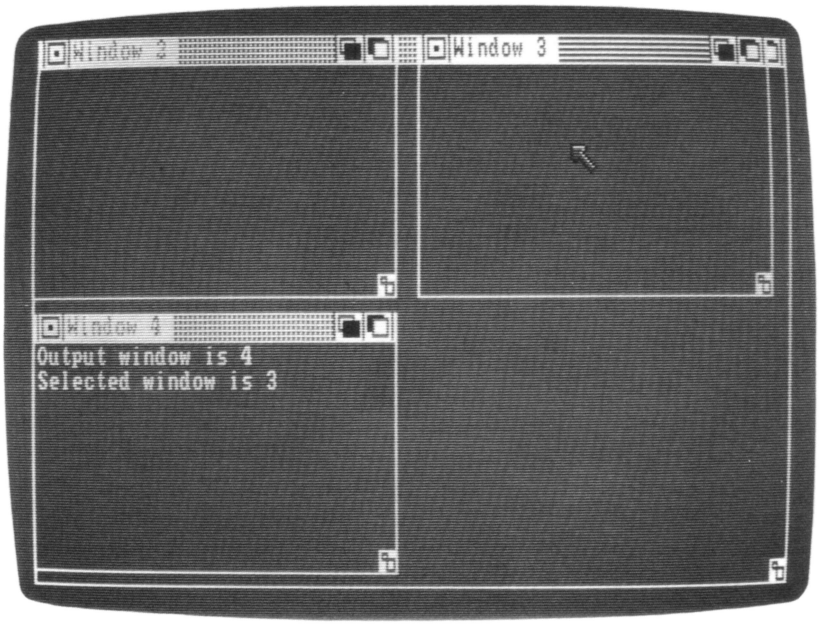


Figura 6-5.

Le funzioni WINDOW(0) e WINDOW(1) indicano la finestra output e quella selezionata.

Il prossimo programma utilizza WINDOW(2) e WINDOW(3), per restituire i valori relativi alle dimensioni della finestra output, e WINDOW(6) per restituire il numero massimo di colori disponibili nella finestra. Le prime 12 linee del programma creano uno schermo e una finestra output, poi un'area pattern che rende più interessante la visualizzazione grafica successiva. La tredicesima linea collega il timer alla funzione RND per renderlo veramente casuale. Le ultime 11 linee creano un ciclo

continuo che riempie la finestra output con figure a 18 lati create col comando AREAFILL.

Nelle prime due linee del ciclo, le due funzioni WINDOW(6) forniscono il numero di colori disponibili, modificati dalle funzioni RND e INT allo scopo di ottenere due numeri casuali interi compresi fra 0 e l'ultimo codice-colore disponibile. Il comando COLOR usa quindi i valori ricavati per selezionare casualmente i colori di sfondo e primo piano.

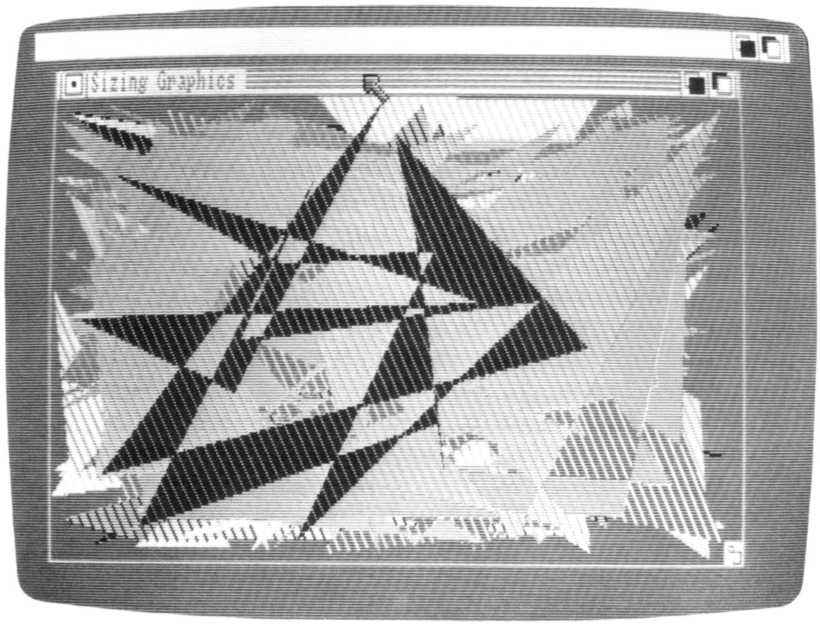
Il ciclo FOR...NEXT che segue fissa 18 punti casuali facendo uso del comando AREA. L'indirizzo casuale dei punti è stabilito utilizzando le funzioni WINDOW(2) e WINDOW(3), in modo da evitare che l'indirizzo capiti al di fuori dei limiti della finestra. Il comando AREAFILL che segue il ciclo FOR...NEXT collega i 18 punti e colora l'area ricavata, mentre il successivo comando GOTO ricomincia il ciclo per creare un'altra figura casuale a 18 lati. La Figura 6-6 è un esempio della grafica casuale ottenuta con questo programma.

Quando fate girare il programma, provate a cambiare le dimensioni della finestra con l'apposito gadget. Quando la finestra cambia misura, le figure riprodotte si adattano di conseguenza.

```
SCREEN 1, 640, 200, 4, 2
WINDOW 2, ''Sizing Graphics '', , 15, 1
DIM area.pat%(7)
area.pat%(0) = &H3F3F
area.pat%(1) = &H9F9F
area.pat%(2) = &HCFCF
area.pat%(3) = &HE7E7
area.pat%(4) = &HF3F3
area.pat%(5) = &HF9F9
area.pat%(6) = &HFCFC
area.pat%(7) = &H7E7E
PATTERN , area.pat%
RANDOMIZE TIMER

Loop:
    forecolor = INT((WINDOW(6) + 1) * RND(1))
    backcolor = INT((WINDOW(6) + 1) * RND(1))
    COLOR forecolor, backcolor
    FOR i = 1 TO 18
        x = INT(WINDOW(2) - 1) * RND(1)
        y = INT(WINDOW(3) - 1) * RND(1)
        AREA(x,y)
    NEXT i
    AREAFILL
    GOTO Loop
```

Figura 6-6.

La funzione WINDOW() permette alle figure della finestra di adattare misure e colori qualora le caratteristiche della finestra cambino.



L'ultimo esempio di funzione WINDOW() è una subroutine che prende una stringa di testo qualsiasi, insieme alle coordinate della posizione in cui volete stamparla, e visualizza il testo in un riquadro. WINDOW(4) e WINDOW(5) forniscono gli indirizzi della locazione iniziale e finale del cursore testo. Il comando LINE racchiude il testo in un riquadro sfruttando gli indirizzi citati e utilizza qualche pixel in più o in meno per contenere tutte le lettere nel box (incluso, ad esempio, il piede della p) e per evitare che le lettere più alte tocchino il bordo superiore del riquadro. Le prime quattro linee del programma definiscono la stringa, stabiliscono l'indirizzo del cursore testo e poi richiamano la subroutine *BoxIt*. Il risultato finale è presentato nella Figura 6-7 (nella pagina seguente).

```
text$ = ''Box it up!''
row = 4: column = 8
GOSUB BoxIt
END

BoxIt:
    LOCATE row, column
    x1 = WINDOW(4): y1 = WINDOW(5)
    PRINT text$;
    x2 = WINDOW(4): y2 = WINDOW(5)
    LINE(x1 - 2,y1 - 7) - (x2 + 1,y2 + 2), 3, B
    RETURN
```

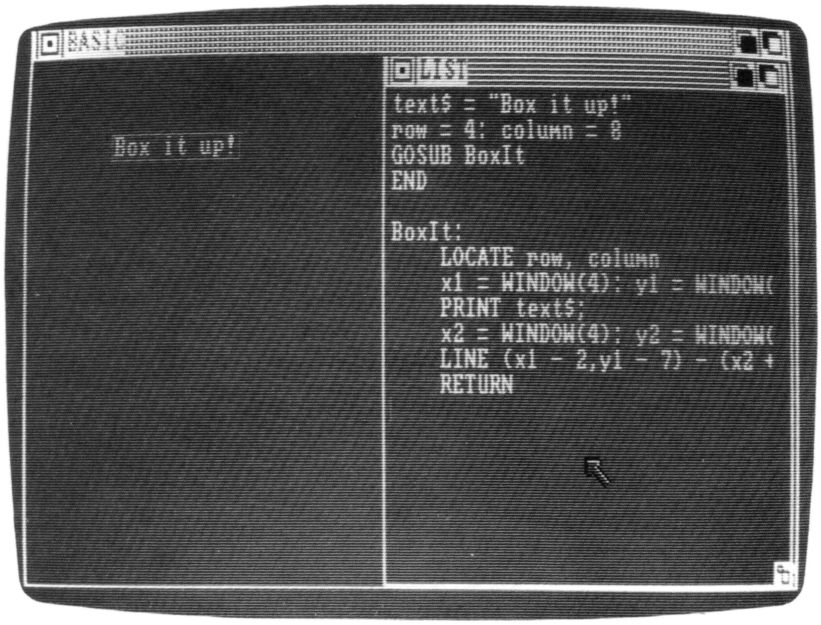


Figura 6-7.

WINDOW(4) e WINDOW(5) localizzano l'inizio e la fine di una stringa consentendo di racchiuderla in un riquadro.

LA FUNZIONE POINT

La funzione POINT controlla il pixel indicatole e fornisce il numero del codice-colore usato per colorare quel pixel. La sintassi è semplice:

POINT *(indirizzo pixel)*

L'indirizzo può essere solo assoluto, non relativo.

Il breve programma riportato nella pagina successiva crea tre aree nella finestra output, ognuna di un colore diverso. Poi, per mezzo di un ciclo FOR...NEXT, posiziona il cursore testo all'interno di ciascuna delle tre aree e utilizza la funzione POINT per leggere il colore del testo e selezionare una tonalità che lo faccia risaltare una volta stampato all'interno dell'area colorata.

```
LINE(0,0) - (620,60), , BF
LINE(0,61) - (620,120), 2, BF
LINE(0,121) - (620,180), 3, BF
FOR i = 0 TO 2
    LOCATE(7 * i) + 3, 5
    background = POINT(WINDOW(4),WINDOW(5))
    COLOR background - 1, background
    PRINT ''Popocatepetl''
NEXT i
```

COME COPIARE E FONDERE LE IMMAGINI GRAFICHE

Potrà capitarvi di voler copiare immagini da una zona a un'altra della finestra output, senza però doverle ridisegnare. L'Amiga BASIC ha due comandi – GET e PUT – che consentono di copiare una sezione quadrangolare della finestra per poi trasferirla in un'altra finestra o in un'altra locazione della finestra in cui già si trova.

IL COMANDO GET

Il comando GET serve per indicare la sezione di finestra, definita blocco grafico, che si desidera copiare. GET serve anche per specificare il nome dell'array necessario a immagazzinare il blocco. La sintassi di GET è:

GET *indirizzi angoli opposti, nome array(numero indice)*

Come già visto nel caso delle opzioni B e BF del comando LINE, GET deve ricevere l'indicazione degli indirizzi degli angoli opposti allo scopo di definire il blocco grafico. Gli indirizzi possono essere solo assoluti, non relativi. Vanno racchiusi fra parentesi e separati da un trattino. Ad esempio, (30,0) - (60,40) definisce un blocco largo 31 pixel e alto 41. Il parametro "nome array" è il nome dell'array utilizzato per immagazzinare il blocco grafico. L'ultimo dato è il numero di indice, facoltativo e spiegato più avanti nel corso di questo capitolo.

Creazione di un array per immagazzinare un blocco grafico

Prima di usare un comando GET, è necessario creare un array che memorizzi i dati grafici del blocco. Per creare l'array, definitene le dimensioni con un comando DIM. Potete usare

variabili di qualsiasi genere, tranne che variabili stringa; in ogni caso, la cosa migliore da fare è utilizzare un array intero breve, specificato semplicemente aggiungendo il segno % alla fine del nome.

Le dimensioni del blocco grafico determinano le dimensioni dell'array; più grande è il blocco, più grande deve essere l'array per contenerlo. Un blocco grafico ha tre dimensioni: larghezza, altezza e profondità. Le prime due dimensioni sono misurate in pixel, la terza, equivalente alla profondità dello schermo su cui il blocco è visualizzato, si misura in bit-plane. Prendete come esempio un blocco grafico che si estende dal pixel (25,3) al pixel (240,76) su uno schermo formato da tre bit-plane. Il blocco grafico misura 216 pixel in larghezza (dalla colonna 25 alla 240 inclusa), 74 pixel in altezza (dalla linea 3 alla 76 compresa) e tre bit-plane in profondità (la profondità dello schermo).

Un breve ripasso degli array vi aiuterà a capire come GET li utilizza per immagazzinare un blocco grafico. Quando create un array con un comando DIM, non fate altro che stabilire il numero di elementi che compongono l'array. Per esempio,

```
DIM graphics%(34)
```

dimensiona un array chiamato *graphics%* di 35 elementi, numerati da 0 a 34. Ogni elemento dell'array è una variabile in grado di memorizzare dati, specificata dal numero di indice posto fra parentesi. Ad esempio, *graphics%(0)* è il primo elemento dell'array, *graphics%(1)* il secondo, *graphics%(2)* il terzo, e così via.

Come gli array immagazzinano dati grafici

L'array intero utilizzato per memorizzare un blocco grafico usa due byte di RAM per il contenuto di ogni elemento dell'array. Ciascun elemento può memorizzare un numero qualsiasi compreso fra −32768 e +32767, oppure dati grafici a gruppi di 16 bit assegnati dal comando GET. Quando GET ricorre all'array per memorizzare il blocco grafico, utilizza i primi tre elementi (0, 1 e 2) per memorizzare le dimensioni del blocco. Il primo elemento dell'array immagazzina la larghezza, il secondo l'altezza e il terzo la profondità.

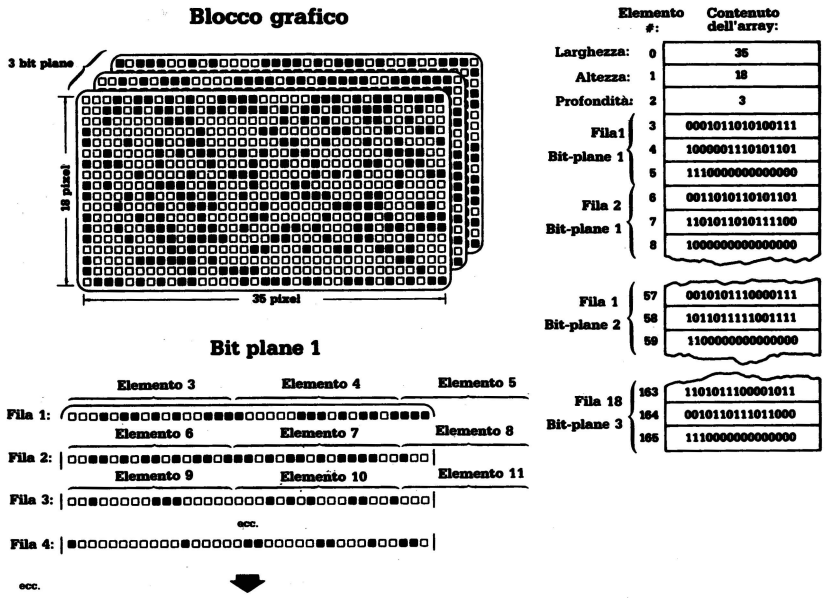
Il resto dell'array memorizza i dati del blocco grafico. I dati vengono memorizzati per bit-plane, iniziando con la prima fila di bit del primo bit-plane, proseguendo fino all'ultima fila del primo bit-plane e ripetendo l'operazione per tutti i bit-plane che seguono. La Figura 6-8 presenta la tecnica di immagazzinamento dei bit-plane all'interno degli elementi dell'array.

È interessante notare come GET non interrompa alcun elemento dell'array al termine di una riga di bit-plane. Se non ci

sono bit sufficienti a riempire completamente l'ultimo elemento della linea, esso viene riempito quanto possibile e poi completato con l'aggiunta di zeri. Ad esempio, per memorizzare un blocco largo 35 pixel sono necessari tre elementi. I primi due elementi dell'array immagazzinano 32 bit, il terzo 3 bit; gli altri 13 bit sono marcati come zeri.

Figura 6-8.

Tecnica di immagazzinamento dei dati di un blocco grafico in un array.



Come calcolare le dimensioni dell'array

Per calcolare quali dimensioni deve avere l'array per memorizzare un blocco grafico, calcolate prima quanti elementi vi servono per memorizzare una linea bit-plane del blocco. Per ottenere il risultato è sufficiente dividere la larghezza in pixel del blocco per 16, ricavando il numero di pixel che sarà contenuto in ogni elemento dell'array. Se avanza qualche pixel, aggiungete un elemento al risultato ottenuto. Per esempio, se avete un blocco largo 35 pixel, l'array deve avere 3 elementi: 35 diviso 16 dà come risultato 2 col resto di 3. I tre pixel di resto occuperanno comunque (come già visto) un intero elemento, per cui il numero totale di elementi necessari è 3.

Fatta questa prima operazione, moltiplicate gli elementi ottenuti per l'altezza e per la profondità del blocco grafico. Il risultato corrisponde al numero di elementi necessari a immagazzinare i dati grafici dell'intero blocco. Aggiungete altri tre elementi per le tre dimensioni e avrete la dimensione finale dell'array adatto al vostro blocco.

Facciamo un esempio. Se dovete memorizzare un blocco grafico che si estende dalla locazione (36,5) alla (105,72) in uno schermo a 16 colori (4 bit-plane), avete una larghezza di 70

pixel, un'altezza di 68 pixel e una profondità di 4 bit-plane. Per immagazzinare una linea sono necessari 5 elementi dell'array (visto che 70 diviso 16 fa 4 col resto di 6). Moltiplicando 5 per l'altezza (68) e la profondità (4) si ottiene 1360. Aggiungete 3 per le dimensioni del blocco e otterrete un totale di 1363 elementi. Utilizzate quindi il comando

```
DIM block%(1362)
```

per creare un array intero col nome di *block%*, formato da 1363 elementi. Dato che ogni array inizia con l'elemento 0, per ottenere 1363 elementi bisogna dimensionare l'array col numero 1362.

Immagazzinamento del blocco grafico nell'array

Una volta creato, l'array va fatto seguire da un comando GET che copi il blocco grafico nell'array. Basta immettere gli indirizzi degli angoli opposti seguiti da una virgola e dal nome dell'array, in questo modo:

```
GET(36,5) - (105,72), block%
```

Copia di più blocchi delle stesse dimensioni in un unico array

Se volete copiare diversi blocchi delle stesse dimensioni, potete risparmiarvi la fatica di creare un array per ogni blocco, semplicemente immagazzinando tutti i blocchi in un unico array. Quando dimensionate l'array, calcolate dapprima la quantità di elementi necessari a contenere un blocco, col metodo appena visto. Poi dimensionate un array come array bidimensionale, utilizzando come primo parametro le dimensoni di un blocco e come secondo parametro il numero dei blocchi che volete memorizzare.

Ad esempio, per copiare cinque blocchi della stessa misura di quello usato nell'ultimo esempio, dovete creare un array intero bidimensionale che misuri 1363 elementi per 5 blocchi, operazione svolta col seguente comando:

```
DIM block%(1362,4)
```

Ricordate che dal momento che ogni numero di indice inizia a 0, *block%* misura 1363 elementi per 5 blocchi anche nel caso in cui i numeri DIM siano inferiori di uno.

Quando salvate i cinque blocchi grafici con cinque comandi GET, i numeri di indice variano per ogni blocco. Il primo blocco viene memorizzato partendo da *block%(0,0)* e finendo a *block%(1362,0)*. Il secondo blocco inzia a *block%(0,1)* e termina con *block%(1362,1)*. E così via fino al quinto blocco, che verrà immagazzinato partendo da *block%(0,4)* e terminando a *block%(1362,4)*.

Per portarsi all'inizio dei dati di ogni blocco grafico, bisogna specificare il primo numero dell'array come 0 e il secondo come numero del blocco relativo (da 0 a 4). Risulta così molto facile utilizzare diversi comandi GET per immagazzinare blocchi usando la stessa variabile. Ad esempio, i cinque comandi GET riportati qui di seguito memorizzano cinque differenti blocchi grafici in cinque diverse parti dell'array block%:

```
GET(36,5) - (105,72), block%(0,0)
GET(46,75) - (115,142), block%(0,1)
GET(136,5) - (205,72), block%(0,2)
GET(236,5) - (305,72), block%(0,3)
GET(36,105) - (105,72), block%(0,4)
```

Copia di più blocchi di diverse dimensioni in un unico array

Se volete immagazzinare più blocchi grafici di misure differenti in un unico array, dovete usare un array a una dimensione, come se si trattasse di salvare un singolo blocco grafico. Per dimensionare l'array, calcolate innanzitutto il numero di elementi necessario per ogni blocco, usando sempre il sistema già visto. Poi sommate tutti i valori per ottenere il numero di elementi necessari a memorizzare tutti i blocchi in un solo array. I blocchi vengono immessi nell'array uno dopo l'altro.

Prendete come esempio l'uso di GET per memorizzare tre blocchi differenti di uno schermo formato da 3 bit-plane. Il primo blocco si estende da (0,0) a (158,40), il secondo da (26,75) a (310,146) e il terzo da (50,150) a (74,174). I calcoli degli elementi necessari per l'immagazzinamento daranno come risultato 1233 per il primo blocco, 3891 per il secondo e 153 per il terzo. Sommando tutti gli elementi si ricava che l'array, per contenere tutti i blocchi, deve avere 5277 elementi, per cui scriverete

```
DIM block%(5276)
```

per creare l'array desiderato.

Quando copiate i tre blocchi grafici nell'array, gli elementi dallo 0 al 1232 vengono occupati dal primo blocco, che ha bisogno di 1233 elementi. Il secondo blocco, che occupa 3891 elementi, inizierà al 1233 e finirà al 5123. Dal 5124 al 5276 sarà

infine memorizzato il terzo blocco di 153 elementi. È consigliabile annotare il numero d'inizio di ogni blocco, in modo da poterlo utilizzare con tre diversi comandi GET quando copierete i tre blocchi. Ad esempio, i seguenti comandi salvano i tre blocchi dell'esempio, partendo dal numero 0 per il primo blocco, dal numero 1233 per il secondo e dal numero 5124 per il terzo.

```
GET(0,0) - (158,40), block%(0)
GET(26,75) - (310,146), block%(1233)
GET(50,150) - (74,174), block%(5124)
```

Potete mettere quanti blocchi volete in un array, purché dimensioniate un array sufficientemente grande e non superiate i limiti della memoria.

IL COMANDO PUT

Il comando PUT preleva l'immagine salvata da GET in un array e la colloca nella finestra alla locazione da voi specificata. PUT ha la seguente sintassi:

```
PUT indirizzo, nome array(numero indice), tipo
```

L'indirizzo può essere assoluto o relativo. Il nome dell'array è lo stesso usato col comando GET quando avete memorizzato il blocco grafico. Il parametro numero indice è opzionale, così come il parametro tipo. Quest'ultimo, se usato, deve essere una parola (fra cinque disponibili) che specifica il tipo di operazione da effettuare fra l'immagine dell'array e l'immagine eventualmente presente sullo schermo che potrebbe essere coperta dall'immissione del blocco.

L'indirizzo nel comando PUT

L'indirizzo incluso nel comando PUT determina il punto della finestra in cui apparirà l'angolo superiore sinistro del blocco grafico. Qualsiasi indirizzo è accettato. Se l'indirizzo specificato è tale da far fuoriuscire parte del blocco dalla finestra, il comando PUT elimina tutta la parte di immagine che oltrepassa i limiti della finestra stessa.

Il nome e l'indirizzo dell'array

Quando specificate il nome dell'array nel comando PUT, usate lo stesso nome utilizzato per salvare il blocco con GET. Se con GET non avete usato numeri dopo il nome, non ne avrete bisogno neanche con PUT. Ad esempio, se avete salvato un blocco grafico nell'array *swath%*, il comando

```
PUT(0,0), swath%
```

sarà sufficiente a visualizzare il blocco grafico *swath%* nella finestra, posizionando l'angolo superiore sinistro dell'immagine nella locazione (0,0).

Se invece avete salvato più blocchi in un array utilizzando numeri diversi per contraddistinguere l'inizio di ciascun blocco, dovrete usare con PUT lo stesso nome e anche lo stesso numero immessi col comando GET. Ad esempio, per visualizzare i cinque blocchi delle stesse dimensioni che avete memorizzato in un esempio precedente nell'array *block%*, immettete questi cinque comandi PUT:

```
PUT(0,0), block%(0,0)
PUT(80,0), block%(0,1)
PUT(160,0), block%(0,2)
PUT(240,0), block%(0,3)
PUT(0,100), block%(0,4)
```

È anche possibile copiare lo stesso blocco più volte in posizioni differenti. È sufficente usare sempre lo stesso nome di array e gli stessi numeri. I quattro comandi seguenti copiano il blocco memorizzato nell'array *swath%* in quattro punti diversi dello schermo:

```
PUT(0,0), swath%
PUT(50,50), swath%
PUT(110,100), swath%
PUT(204,0), swath%
```

Il parametro tipo

Ci sono cinque diverse operazioni che potete specificare nel comando PUT per stabilire in che modo l'immagine immessa col blocco debba fondersi con l'immagine già presente sullo schermo.

- **PSET**
  Inserisce il blocco grafico nella finestra coprendo tutto quello che c'è sullo schermo in quel momento.

- **PRESET**
  Immette il blocco grafico nella finestra come fa PSET, ma prima cambia tutti i colori con i rispettivi colori inversi.

- **AND**
  Esegue l'operazione AND dell'algebra di Boole fra i pixel del blocco e i pixel della finestra output che devono essere coperti, visualizzando poi i risultati. Per capire cosa succede osservate attentamente i risultati e tralasciate il

procedimento. Dovunque sia presente un colore di sfondo nella finestra output (codice-colore 0), AND non depone alcuno dei contenuti del blocco grafico. Dovunque sia presente l'ultimo colore possibile della finestra (il codice-colore più alto possibile), AND inserisce i colori del blocco grafico così come sono. Gli altri colori si combinano con i blocchi grafici, dando vita a colori che si possono prevedere solo con l'ausilio dell'algebra di Boole (AND).

- **OR**
  Esegue l'operazione OR dell'algebra di Boole fra i pixel del blocco grafico e i pixel della finestra output che devono essere coperti. OR è l'opposto di AND: dove trova il colore di sfondo inserisce i pixel del blocco grafico nei suoi colori originari, mentre lascia tutto invariato dove è presente l'ultimo colore possibile della finestra (il codice colore più alto). Ogni altro colore del blocco si combina con quelli della finestra per creare nuovi colori che si possono prevedere solo con l'ausilio dell'algebra di Boole (OR).

- **XOR**
  Esegue l'operazione XOR dell'algebra di Boole fra i pixel del blocco grafico e i pixel della finestra output che devono essere coperti. XOR inserisce il blocco grafico nei suoi colori originali dovunque un pixel del colore di sfondo nella finestra output sia coperto con un pixel di un blocco grafico. Dovunque nel blocco grafico un colore di sfondo copre un pixel della finestra di output, XOR inserisce il pixel colore della finestra output. Quando si mescolano due colori non di sfondo, il risultato è un nuovo colore.

Se non viene specificata alcuna preferenza, PUT presuppone che la scelta sia XOR.

XOR ha una proprietà molto utile. Se usate XOR per mettere lo stesso blocco nella medesima posizione per due volte, la prima volta il blocco apparirà con i colori determinati dalla scelta di XOR, mentre la seconda volta il blocco posizionato col primo comando XOR sparirà completamente, facendo riapparire lo schermo sottostante così com'era prima del comando PUT. In pratica il secondo XOR serve per annullare il primo.

Le applicazioni di PSET e PRESET sono ovvie: questi comandi possono essere usati per inserire blocchi grafici normali o in colori inversi in qualsiasi momento lo desideriate. Le applicazioni di AND e OR sono un po' meno evidenti. È possibile usarli con una maschera, allo stesso modo in cui un pittore usa uno stampo. Ritagliando la carta si ottengono delle

sagome molto utili per dipingere solo l'area desiderata, essendo la zona circostante protetta dalla carta. Allo stesso modo, una maschera per lo sfondo definisce una sagoma nella finestra che sarà poi usata dal comando PUT quando immetterà il blocco grafico utilizzando le opzioni AND o OR.

Per creare una maschera, dovete definire una sagoma nella finestra utilizzando il codice-colore più elevato che avete a disposizione e poi circondarla col colore dello sfondo. Quando inserite un blocco grafico usando l'opzione AND, la maschera sarà riempita dai pixel del blocco mentre l'area circostante resterà immutata. Il risultato con l'opzione OR è esattamente l'opposto: OR lascia intatta la maschera e riempie l'area circostante coi pixel del blocco. Potrete vedere un esempio di maschera in un programma riportato più avanti in questo capitolo.

Per constatare l'effetto delle cinque opzioni osservate la Figura 6-9 (riportata a pagina 186).

UN PROGRAMMA DIMOSTRATIVO CHE USA GET E PUT

Il programma riportato nella pagina seguente salva cinque blocchi grafici diversi in un array a due dimensioni e poi li immette sullo schermo in posizioni diverse usando tutte e cinque le opzioni. Le prime undici linee di programma creano una finestra in uno schermo formato da 4 bit-plane, poi utilizzano due cicli FOR...NEXT per riempire la parte superiore della finestra con parole e sfondo colorati con tutte le tonalità consentite dalla finestra. Le otto linee successive stampano parti di testo e disegnano cerchi nella metà inferiore dello schermo, utilizzando il codice-colore più elevato possibile (numero 15, grigio chiaro).

Il comando DIM della linea successiva dimensiona un array intero a due dimensioni, composto da 803 elementi per 5. Questo array serve a memorizzare cinque blocchi, ognuno dei quali ha bisogno di uno spazio di 803 elementi. Il ciclo FOR...NEXT che segue utilizza cinque volte il comando GET per immagazzinare un blocco della sezione superiore dello schermo. Il comando LINE delimita l'area salvata per mostrarla anche all'operatore. L'altro ciclo FOR...NEXT blocca brevemente la visualizzazione, per dare modo di vedere come si presenta lo schermo prima che i blocchi grafici vengano reinseriti nella finestra.

Le ultime cinque linee sono comandi PUT che immettono i cinque blocchi nella parte inferiore dello schermo, uno in fila all'altro. Ogni comando PUT specifica un'opzione diversa.

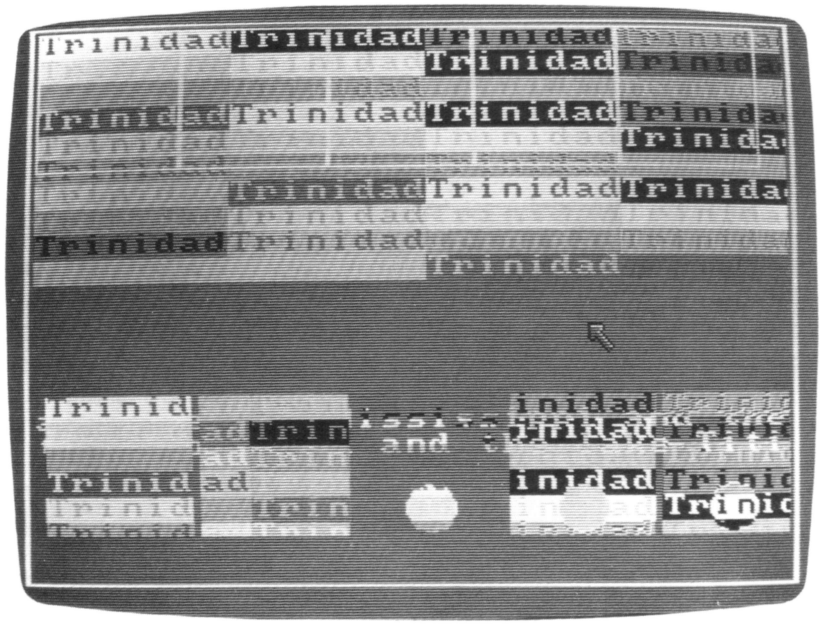
```
SCREEN 1, 320, 200, 4, 1
WINDOW 2, , , 0, 1
WIDTH 40
FOR i = 1 TO 3
    FOR j = 0 TO 15
        k = j + 1
        IF k > 15 THEN k = 0
        COLOR j, k
        PRINT ''Trinidad'';
    NEXT j
NEXT i
LOCATE 16,1
PRINT ''and the big Mississippi and the'';
PRINT ''town Honolulu and the lake'';
PRINT ''Titicaca.''
FOR i = 35 TO 287 STEP 63
    CIRCLE(i,170), 10
    PAINT(i,170)
NEXT i
DIM block%(802,4)
FOR i = 0 TO 4
    GET(60 * i,0) - ((60 * i) + 59,49), block%(0,i)
    LINE(60 * i,0) - ((60 * i) + 59,49), 1, B
NEXT i
FOR t = 1 TO 5000: NEXT t
PUT(5,130), block%(0,0), PSET
PUT(68,130), block%(0,0), PRESET
PUT(131,130), block%(0,0), AND
PUT(194,130), block%(0,0), OR
PUT(257,130), block%(0,0), XOR
```

I risultati sono presentati nella Figura 6-9 (riportata nella pagina successiva).

Notate il modo in cui le opzioni influiscono sui cinque blocchi reinseriti sullo schermo. Il primo blocco, che utilizza PSET, è copiato direttamente nella finestra in modo che copra qualsiasi cosa sia presente nella sua posizione. Il secondo blocco, che usa PRESET, subisce la stessa operazione ma viene stampato con i colori inversi. Nel terzo blocco, le lettere e il cerchio fungono da maschera poiché sono colorati col registro più alto disponibile, il numero 15; il blocco viene riprodotto esclusivamente entro i limiti dei caratteri e del cerchio, ma non all'esterno sul colore di sfondo. Nel quarto blocco succede esattamente l'opposto: il blocco viene riprodotto sui pixel dello sfondo mentre i caratteri e il cerchio restano immutati. Nell'ultimo blocco, XOR trasforma i caratteri in colori totalmente nuovi, diversi a seconda dei colori che si sovrappongono.

Figura 6-9.

I comandi GET e PUT copiano blocchi da una sezione all'altra della finestra utilizzando le cinque opzioni disponibili.

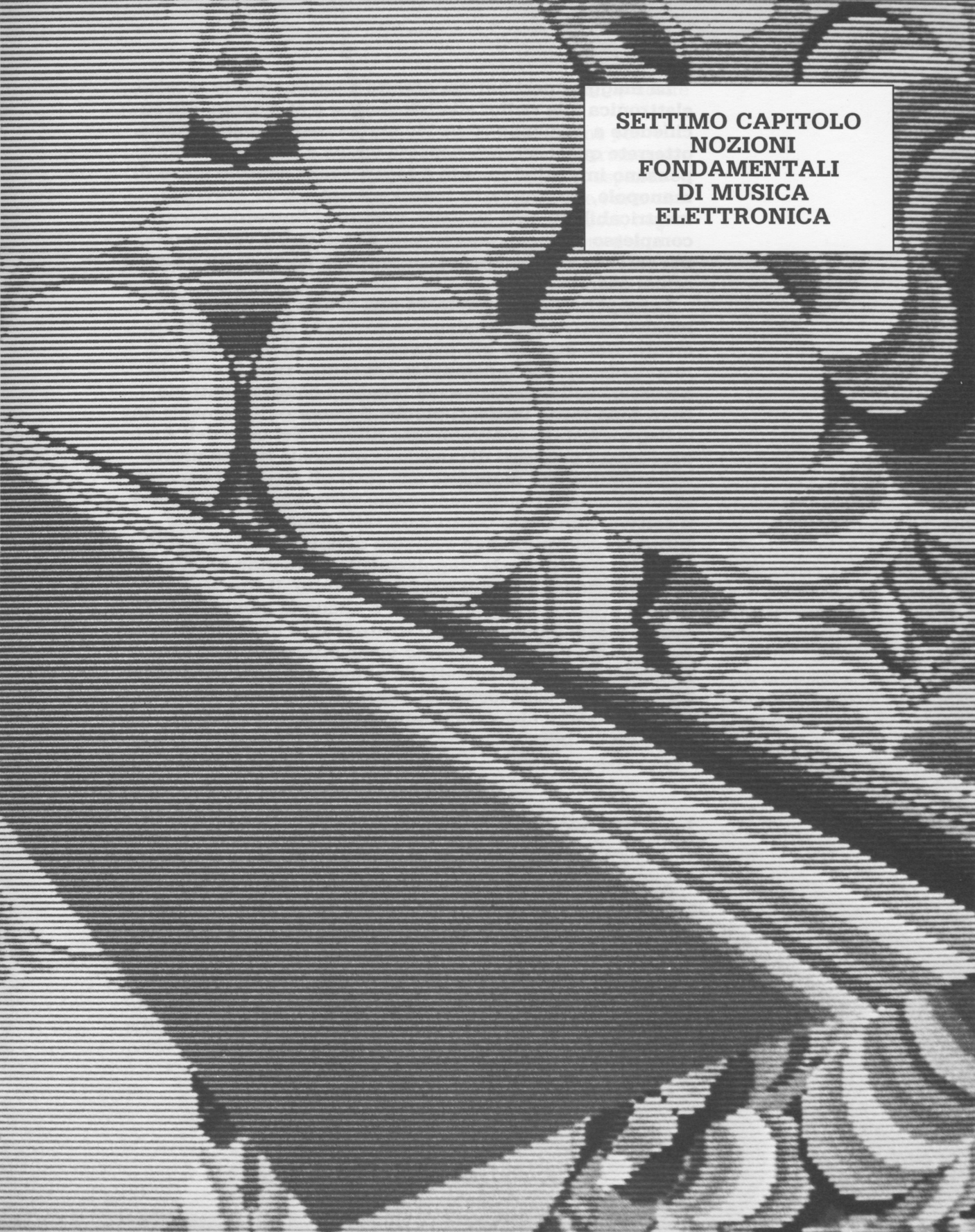


Avete così provato a usare anche i comandi grafici meno comuni dell'Amiga BASIC, e avete aggiunto parti di testo alle vostre creazioni grafiche utilizzando fonti differenti e colori svariati. Abbiamo parlato anche della grafica interattiva che si adatta alle mutevoli condizioni delle finestre. Abbiamo infine spiegato come sia possibile, con i comandi GET e PUT, copiare blocchi grafici da una sezione all'altra dello schermo, dopo averli immagazzinati in un array. Tutte queste informazioni si riveleranno utili quando utilizzerete PUT per creare sequenze animate. Ma tutto ciò sarà spiegato solo nella sezione 4. Per ora pensate a leggere la prossima sezione, dedicata alle capacità sonore dell'Amiga.

SEZIONE 3

Suoni

In questa sezione imparerete come creare musica e sintesi vocale con l'Amiga.
Il settimo capitolo espone i principi fondamentali della musica elettronica.
Nell'ottavo capitolo vedremo come utilizzare al meglio le possibilità offerte da Deluxe Music, esaminando anche altri programmi ed elementi hardware che sviluppano ulteriormente le già ottime qualità sonore dell'Amiga.
Il nono capitolo illustra infine i comandi del BASIC necessari per riprodurre musica, effetti sonori e voce umana.



SETTIMO CAPITOLO
NOZIONI FONDAMENTALI DI MUSICA ELETTRONICA

La maggior parte di noi ascolta continuamente musica elettronica alla radio, alla televisione e al cinema. Ma se chiedete a qualcuno come sia stata creata quella musica, otterrete quasi sempre risposte vaghe e imprecise. Molti pensano immediatamente a gigantesche console piene di manopole, interruttori e lucine colorate, a un groviglio inestricabile di cavi elettrici, oppure si immaginano un complesso rock con enormi pile di tastiere e cavi sparsi per tutto lo studio. Ma tutto questo fa parte di un passato ormai oscuro.

I computer hanno impresso una svolta decisiva alla musica elettronica, rendendola semplice e divertente da creare. Non servono più banconi stipati di interruttori e manopole, perché il computer concentra tutti i controlli necessari nella tastiera e nel mouse. I collegamenti via cavo sono meno intricati perché il numero di elementi da collegare è stato notevolmante ridotto. Nel caso dell'Amiga, l'intero processo di creazione musicale risulta piuttosto semplice, dal momento che tutti gli elementi necessari sono contenuti nella macchina: tutto quello che serve è il software adatto a controllare tali elementi.

Dato che l'Amiga può svolgere diverse funzioni audio, è opportuno affrontarle singolarmente, cercando di capire come funzionano e qual è il loro compito. Il fine dei magici effetti audio dell'Amiga è quello di compiacere il vostro udito; a questo scopo è altrettanto utile capire il modo in cui l'orecchio umano viene stimolato dalla percezione dei suoni provenienti dal mondo esterno. Più vasta sarà la vostra conoscenza in materia, più incredibili saranno i suoni che riuscirete a produrre con l'Amiga.

LA PERCEZIONE DEI SUONI

Quasi tutti sanno che il suono giunge al nostro udito propagandosi attraverso l'aria. Infatti, tappandosi le orecchie con le mani, si riesce a escludere gran parte dei suoni circostanti. È altrettanto evidente che le fonti da cui provengono i suoni sono molto varie. In una stanza piena di gente che parla, è facile riconoscere ciò che sta dicendo la persona con cui state conversando (purché le siate sufficientemente vicini). Quello che molti non sanno è come tutto questo sia possibile.

Perché alcuni suoni sono diversi da altri? Come fanno le nostre orecchie a interpretare ciascun suono?

I suoni hanno origine da un oggetto vibrante, come per esempio un gong, le corde vocali o la corda di uno strumento musicale. Nella vibrazione, l'oggetto in questione si espande dapprima verso l'esterno, comprimendo leggermente l'aria circostante. Viceversa, quando si contrae in senso opposto, l'oggetto crea uno spazio di aria rarefatta, quasi un vuoto d'aria. Dal momento che la vibrazione si protrae per un certo tempo, si viene a creare nell'aria tutta una serie di compressioni e vuoti. Queste variazioni di pressione si dipartono dall'oggetto sotto forma di onde sonore, come illustrato dalla Figura 7-1, e si diffondono nell'aria con lo stesso movimento concentrico delle increspature causate da una pietra gettata in acqua. Le onde sonore viaggiano finché si affievoliscono completamente o finché incontrano altri oggetti (non ultimo l'orecchio umano).

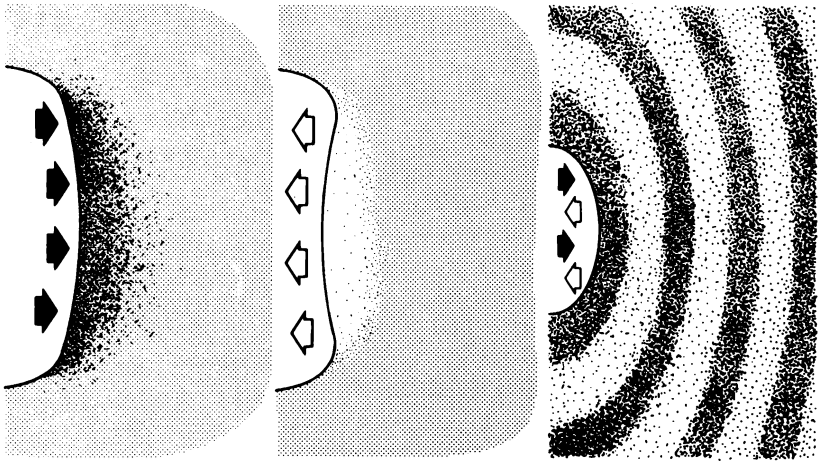


Figura 7-1.

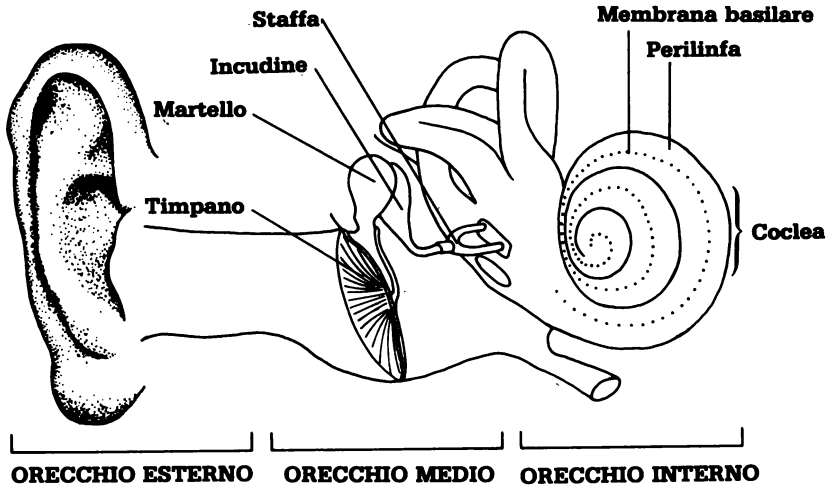
Nel disegno a sinistra, un oggetto vibra proiettandosi verso l'esterno e comprimendo l'aria circostante. Nel riquadro centrale, lo stesso oggetto vibrante si ritrae, rendendo l'aria circostante più rarefatta. Il disegno di destra mostra una serie di queste compressioni e rarefazioni che si diffondono nell'aria sotto forma di onde sonore.

LA STRUTTURA DELL'ORECCHIO

L'orecchio umano è formato da tre parti: l'orecchio esterno, cioè la parte che sporge dalla testa, l'orecchio medio e l'orecchio interno (vedere Figura 7-2 nella pagina successiva). L'orecchio esterno, carnoso e di forma involuta, raccoglie e concentra le onde sonore, facendole pervenire all'orecchio medio. È qui che le onde incontrano il timpano, una sottile membrana che trasmette le vibrazioni a tre ossicini (martello, incudine e staffa), i quali a loro volta amplificano le onde sonore e le trasmettono alla coclea (o chiocciola), che fa già parte dell'orecchio interno.

Figura 7-2.

Le tre parti dell'orecchio.



La coclea è una cavità della parte ossea del cranio, di forma simile al guscio di una lumaca, piena di un liquido detto perilinfa e divisa in senso longitudinale da una membrana molto sensibile, chiamata membrana basilare. La vibrazioni amplificate dal timpano vengono trasmesse al liquido perilinfatico che a sua volta fa vibrare la membrana basilare. Le terminazioni nervose della membrana trasmettono la percezione delle vibrazioni al cervello. Il cervello analizza le vibrazioni consentendo così la percezione del suono.

LE CARATTERISTICHE DEL SUONO

Nell'analisi che fa del suono, il cervello distingue quattro caratteristiche fondamentali delle vibrazioni: frequenza, ampiezza, timbro e durata. Queste quattro caratteristiche, che consentono di descrivere qualsiasi suono udibile dall'uomo, possono essere utilizzate come parametri per la creazione dei suoni desiderati.

Frequenza

La frequenza è data dalla velocità di vibrazione. Se un oggetto vibra lentamente, si genera un suono a bassa frequenza; se le vibrazioni sono invece più veloci, il suono generato sarà ad alta frequenza. La frequenza di un suono ne determina il tono: più alta è la frequenza, più alto sarà il tono.

Come esempio di frequenze diverse, pensate alle corde di una chitarra. Le corde più grosse, essendo più pesanti di quelle sottili, vibrano più lentamente ed emettono di conseguenza un suono di tono più basso.

La frequenza viene misurata in hertz (abbreviato in Hz), un'unità di misura che valuta le vibrazioni al secondo. L'udito umano percepisce mediamente tutti i suoni compresi fra i 20 e i 20.000 Hz.

Ampiezza

L'ampiezza è la forza della vibrazione. Quando un oggetto vibra con forza (con contrazioni che coprono uno spazio relativamente grande), si genera un suono di grande ampiezza. Quando le vibrazioni sono più deboli (le contrazioni dell'oggetto sono quindi limitate nello spazio), si genera un suono di ampiezza inferiore. L'ampiezza di un suono ne determina il volume: più grande è l'ampiezza, più alto è il volume. Pensate ancora all'esempio della chitarra. Se pizzicate una corda con forza, questa vibra notevolmente e produce un suono molto ampio. Pizzicandola con più delicatezza, la corda vibra invece in maniera impercettibile, generando un suono di ampiezza ridottissima.

L'ampiezza di un suono si misura in decibel. Un decibel rappresenta la più piccola variazione di volume percepibile dall'orecchio umano. Aumentando di dieci decibel l'ampiezza di un suono, si ottiene il risultato di raddoppiarne il volume. L'udito umano percepisce i suoni compresi fra un decibel (limite minimo di percezione) e poco più di 120 decibel (limite massimo di sopportazione fisica).

Timbro

Il timbro è un po' più complesso della frequenza o dell'ampiezza: è un insieme di frequenze all'interno di un unico suono. La maggior parte degli oggetti non vibra a una sola frequenza, bensì a più frequenze simultaneamente. La frequenza più bassa è detta fondamentale, ed è la vibrazione che viene percepita come frequenza principale del suono. La fondamentale determina la tonalità del suono. Le frequenze più alte sono chiamate sopratoni; esse si fondono alla frequenza fondamentale variando la qualità tonica del suono.

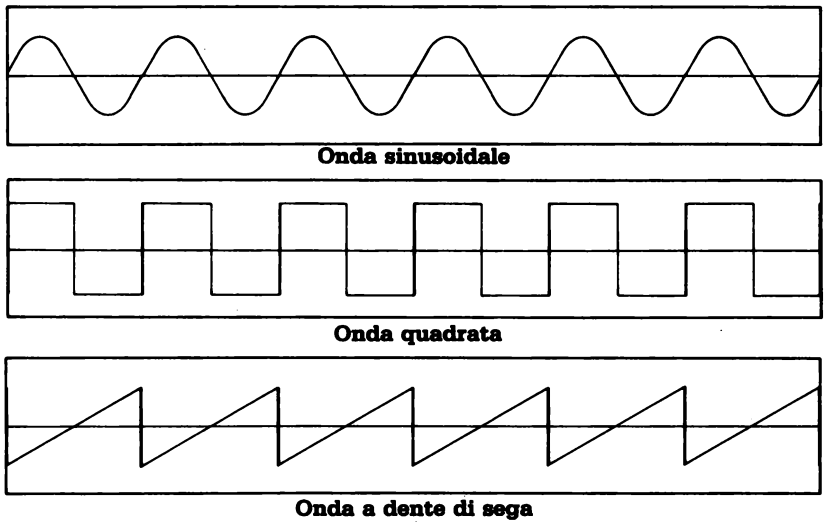
Il timbro di un suono ne determina la scala cromatica. Più è elevato il numero di sopratoni presente in un suono, più ricco risulta il suo timbro; viceversa, meno sono i sopratoni, più sottile è il timbro. Non esistono unità di misura per valutare il timbro, anche se è possibile stabilire il numero esatto di sopratoni presenti in un suono (processo definito "analisi di Fourier"). Ma in genere si preferisce descrivere la qualità del timbro con termini come "ricco", "sottile", "grosso" o "fricativo".

Un buon esempio di timbri differenti è dato da un violino e un flauto che emettono esattamente la stessa nota. Ciascuno dei due strumenti produce suoni che utilizzano gamme di sopratoni completamente diverse; così ognuno di essi possiede un timbro caratteristico, che consente di distinguerlo dagli altri strumenti.

L'analisi di Fourier consente di individuare tutti i sopratoni di un suono, ma il modo più comune di rappresentare graficamente il timbro di un suono consiste nel tracciare una forma d'onda. La forma d'onda è una rappresentazione delle pressioni atmosferiche di un'onda sonora che mette in evidenza la risultante dei sopratoni del suono. Esaminando l'andamento della forma d'onda, è possibile distinguere i vari timbri. La Figura 7-3 riporta le forme d'onda fondamentali: onda sinusoidale, onda quadrata e onda a dente di sega. L'onda sinusoidale rappresenta un suono melodioso e fluido, l'onda quadrata un suono ricco e pieno, e l'onda a dente di sega un suono penetrante e cromatico.

Figura 7-3.

Tre forme d'onda: sinusoidale, quadrata, a dente di sega.



Durata

La durata permette di misurare i primi tre attributi del suono: frequenza, ampiezza e timbro. Usata nel modo più semplice, la durata misura le tre caratteristiche congiuntamente, calcolando la lunghezza dell'intero suono dalla prima percezione all'estinzione totale. In altri casi la durata misura ogni caratteristica singolarmente, registrandone le variazioni nel

tempo. Ad esempio, un cantante può mantenere a lungo la stessa nota, ma variare il volume a cui la emette; la durata permette in questo caso di calcolare gli aumenti e le diminuzioni di ampiezza. Analogamente, un suonatore di tromba può prolungare il suono di una nota, inserendo però a un certo punto la sordina; qui la durata può essere utilizzata per misurare la lunghezza del timbro emesso dalla tromba prima che fosse frapposta la sordina a produrre un timbro più smorzato. Se un sassofonista effettua delle variazioni su una nota, è possibile calcolare il tempo necessario alla nota modificata per riportarsi sulla frequenza della nota finale.

La rappresentazione grafica dell'andamento delle tre caratteristiche nel tempo permette di definirne l'inviluppo. Gli inviluppi sono fondamentali per la descrizione di un suono. Ad esempio, pizzicando una corda d'arpa si genera un rumore molto secco e acuto all'inizio, rumore che sfuma poi nel nulla mentre la nota si protrae. Esaminando l'inviluppo dell'ampiezza della nota in questione (Figura 7-4), si vede come la nota raggiunga subito un'ampiezza notevole (alto volume) per poi affievolirsi col trascorrere del tempo (volume sempre più basso fino all'estinzione totale).

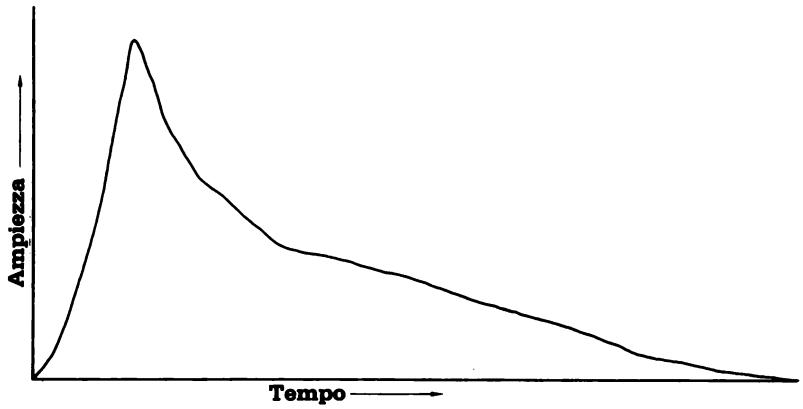


Figura 7-4.

L'inviluppo dell'ampiezza della nota prodotta da un'arpa.

Gli inviluppi sono molto importanti per la percezione sonora. Qualsiasi suono è determinato dalla forma che il suo tono, il suo volume e il suo timbro assumono nel tempo. Variando l'inviluppo di una delle caratteristiche, si determina un cambiamento marcato della natura stessa del suono.

LA PRODUZIONE DI SUONI ELETTRONICI

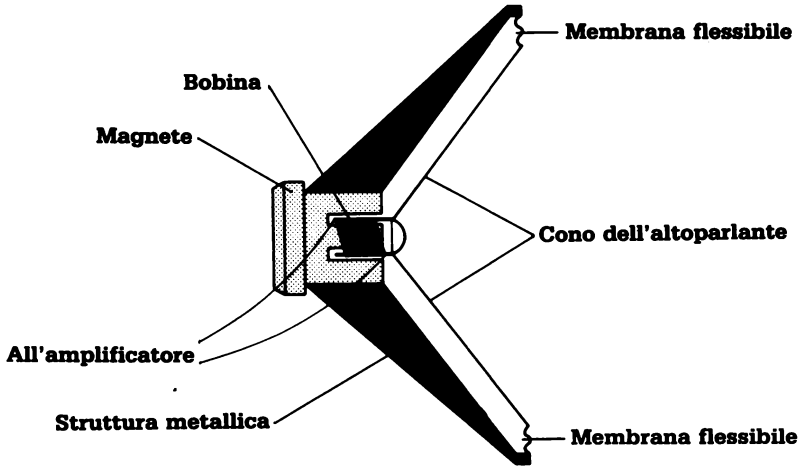
Per percepire i suoni provenienti da una sorgente elettronica, come una piastra di registrazione, un sintetizzatore o un computer, è necessario un sistema formato almeno da due componenti: un altoparlante e un amplificatore che lo controlli. L'amplificatore riceve un debole segnale audio da uno strumento elettronico, lo amplifica e lo invia all'altoparlante, che compie la funzione strettamente fisica di convertire il segnale audio amplificato in un segnale percepibile dall'udito. L'altoparlante ha il difficile compito di creare un'enorme varietà di suoni, che hanno timbri diversi, e coprono una vasta gamma di tonalità e volumi.

L'ALTOPARLANTE

La Figura 7-5 riproduce un altoparlante visto in sezione. La parte che vibra e che produce i suoni è il cono. Il cono dell'altoparlante non è altro che un pezzo conico di carta rigida o di plastica, montato in una cornice metallica circolare. La punta del cono è sospesa all'interno della cornice per mezzo di membrane altamente flessibili, in gomma o simili, che permettono al cono di muoversi all'interno della struttura metallica.

Figura 7-5.

Sezione di un altoparlante.



In un altoparlante classico, il centro del cono è collegato a una bobina di filo elettrico sospesa nel campo magnetico di un magnete molto potente, che è a sua volta attaccato alla parte posteriore della cornice metallica. Questo magnete, fra l'altro, è sufficiente a cancellare il contenuto di un dischetto magnetico; fate quindi attenzione a non mettere i vostri dischi troppo vicini a un altoparlante. I due capi del filo della bobina sono collegati all'amplificatore.

Quando l'amplificatore invia un segnale audio attraverso la bobina, in essa si crea un campo magnetico che attrae oppure respinge il campo creato dal magnete fisso. Di conseguenza, la bobina viene spinta in avanti o all'indietro, trascinando con sé il cono. La direzione e l'intensità dei movimenti del cono dipendono dalla forza del campo magnetico creato dal segnale audio che passa attraverso la bobina: più forte è il campo, più si muoverà il cono.

Il segnale audio proveniente dall'amplificatore cambia la forza del campo magnetico molto rapidamente, e il cono, perciò, vibra abbastanza velocemente da produrre suoni. Quando il segnale audio muta in frequenza, ampiezza e sopratoni, l'altoparlante percepisce i cambiamenti, emettendo un suono che corrisponde sempre al segnale inviato. Un cono di buona qualità interpreterà accuratamente anche la minima variazione del segnale audio, riproducendola nel suono che emette.

In un buon altoparlante ad alta fedeltà ci sono generalmente tre coni, uno per ciascuna delle tre differenti gamme di frequenza. Un cono, detto woofer, produce i suoni bassi. Il cono medio produce suoni di media frequenza e il tweeter è il cono che produce i suoni più acuti. Un sistema di filtri, chiamato rete di crossover (o rete incrociata), divide il segnale audio proveniente dall'amplificatore in tre segnali distinti, uno per ogni tipo di frequenza. I segnali vengono poi inviati all'altoparlante attraverso cavi separati. Il risultato, se i componenti dell'altoparlante sono di buona qualità, è un'eccellente riproduzione di tutte le frequenze percepibili dall'orecchio umano.

L'AMPLIFICATORE

Un segnale audio proveniente da registratori, piatti stereo, sintonizzatori o dallo stesso Amiga, è troppo debole per controllare un altoparlante; per essere percepito, il segnale deve prima essere amplificato. Il compito dell'amplificatore è proprio quello di prelevare il segnale audio di una sorgente molto debole e potenziarlo affinché riesca a muovere il cono nel modo opportuno. Quasi tutti gli amplificatori hanno una manopola per regolare il volume, consentendo all'ascoltatore di controllare il livello di amplificazione del segnale. Alcuni amplificatori dispongono anche della regolazione per i toni, che permette di cambiare il timbro del segnale in entrata prima che venga amplificato e inviato all'esterno.

STEREOFONIA

Chi ascolta musica possiede generalmente un impianto stereo con due altoparlanti o casse, una per la parte destra e una per la parte sinistra dell'area di ascolto. Le casse convertono due segnali audio ben distinti (chiamati canale destro e canale sinistro) in due fonti sonore, le quali danno all'ascoltatore l'illusione che nella stanza vi siano più sorgenti sonore (questo processo è definito imaging).

Per capire come funziona questo meccanismo, considerate il seguente esempio. Un ingegnere del suono deve registrare un trio di fiati disposto sul palco in questo modo: l'oboe a sinistra, il clarinetto al centro e il fagotto a destra. Per la registrazione utilizza due microfoni, uno per ogni canale. Il microfono di sinistra è vicino all'oboe e più lontano dagli altri strumenti, cosicché registra un suono ben distinto di oboe, un suono medio di clarinetto e un suono alquanto debole proveniente dal lontano fagotto. Il microfono di destra, situato vicino al fagotto, registrerà nel canale sinistro un ottimo segnale del suono del fagotto, un suono ancora medio per il clarinetto e un suono debole per l'oboe.

Quando l'ascoltatore sente la registrazione (tenendo la cassa del canale di sinistra alla sua sinistra e la cassa di destra alla sua destra), percepisce gran parte del suono dell'oboe attraverso la cassa sinistra, mentre dalla cassa destra lo strumento è appena percettibile; l'oboe si trova dunque alla sinistra dell'ascoltatore. Il suono del clarinetto si equivale nelle due casse, e la sensazione è che esso provenga dall'area posta al centro delle casse stesse. Il suono del fagotto domina la cassa di destra e quindi l'ascoltatore ne localizza la presenza alla sua destra. Gli impianti stereo usano la tecnica dell'imaging per creare l'illusione di un'orchestra dislocata per tutta la stanza.

L'Amiga dispone di uscite stereo e può essere collegato a un impianto stereofonico, ma i suoi canali destro e sinistro sono nettamente separati: il segnale del canale di sinistra finisce interamente alla cassa sinistra, quello del canale di destra alla cassa destra. I due segnali distinti non consentono di realizzare l'effetto descritto in precedenza, in quanto non esiste una commistione di suoni provenienti da entrambe le casse. Per fondere i segnali sonori dell'Amiga e realizzare l'imaging, è necessario possedere un mixer audio (di cui parleremo più avanti in questo stesso capitolo).

LA SINTESI MUSICALE

Per registrare musica si usano microfoni che trasformano le onde sonore diffuse nell'aria in segnali audio. Quando si ascolta la registrazione, l'impianto sonoro riconverte i segnali audio in onde sonore. *Sintetizzare* musica significa invece creare un segnale audio partendo da zero, cioè non registrare un segnale già definito.

La sintesi musicale richiede strumenti che vanno ben oltre l'amplificatore e l'altoparlante. È necessario un sintetizzatore per costruire il segnale e uno strumento di controllo tipo tastiera o sequencer per suonare. L'Amiga dispone dell'hardware necessario a sintetizzare i suoni attraverso le porte audio. È possibile collegare una tastiera alle porte di controllo o usare direttamente la tastiera dell'Amiga come se fosse un vero e proprio strumento musicale. L'utente può acquistare o creare software che crei un sequencer nella RAM, allo scopo di suonare sequenze di note utilizzando le capacità dell'Amiga.

SINTETIZZATORI

Il concetto di sintetizzatore è alquanto semplice. Il sintetizzatore stabilisce la frequenza, l'ampiezza e la forma d'onda di un segnale audio, inviando poi questo segnale all'amplificatore sotto forma di onda audio non amplificata. L'onda viene poi amplificata e inviata alle casse, che la traducono direttamente in onde sonore, riproducendo in forma udibile l'onda sintetizzata originalmente. Nella pratica la sintesi sonora non è però altrettanto semplice. Ci vuole un grande lavoro per creare un segnale audio di una certa qualità. I primi ad avere successo furono i sintetizzatori analogici, ancor oggi molto popolari.

Sintetizzatori analogici

I sintetizzatori analogici utilizzano una serie di componenti elettronici, in un primo tempo per creare un semplice segnale audio, e poi per alterarlo, rifinirlo e distorcerlo fino a ottenere una forma più complessa e interessante. Il cervello del sintetizzatore analogico è l'oscillatore controllato in tensione, in forma abbreviata VCO. Il VCO, partendo da zero, crea un segnale audio corrispondente a una delle tante forme d'onda disponibili. Queste forme d'onda sono generalmente molto semplici: onde sinusoidali, triangolari, quadrate, a dente di sega e di qualche altro tipo.

Il VCO varia il tono a seconda della tensione, controllata e trasmessa solitamente da una tastiera o da un sequencer. Tastiera o sequencer inviano un voltaggio diverso a seconda della nota, variando così la tonalità del VCO. Per abbellire il suono, il segnale proveniente dal VCO può essere fatto passare attraverso filtri o amplificatori controllati in tensione che aggiungono a esso caratteristiche quali i sopratoni, la durata e l'ampiezza di ogni nota. Il segnale può essere ulteriormente arricchito filtrandolo attraverso altri tipi di sintetizzatori: ring modulator, generatori di rumore bianco, campionatori di suoni, filtri passa-alto, filtri passa-basso e filtri passa-banda. L'effetto che ogni modulo ha sul segnale è diverso, ma tutti hanno lo scopo di trasformare il suono originale, semplice e poco interessante, in un suono più ricco ed evoluto.

Sintetizzatori digitali

L'Amiga, come molti sintetizzatori moderni, è un sintetizzatore digitale. Invece di produrre un segnale semplice per mezzo di un oscillatore, per poi arricchirlo attraverso altre apparecchiature elettroniche, un sintetizzatore digitale crea un suono sotto forma di modello digitale matematico e trasforma poi questo modello in un segnale audio che viene inviato direttamente all'amplificatore. Con i sintetizzatori digitali la creazione di suoni nuovi ed esclusivi diventa una pura questione di matematica oltre che di elettronica.

Per realizzare i suoi modelli, l'Amiga crea nella propria memoria la forma d'onda di un suono. Per capire come funziona questo meccanismo, si pensi al modo in cui una forma d'onda riproduce la qualità di un suono. Consideriamo una forma d'onda come quella della Figura 7-6, letta da sinistra a destra. L'onda sale per indicare una forte pressione dell'aria circostante la fonte sonora, e scende per indicare un abbassa-

mento della pressione. L'andamento ha una diretta corrispondenza con il movimento del cono dell'altoparlante. Quando l'onda supera la linea orizzontale centrale, il cono si proietta verso l'esterno aumentando la pressione. Quando l'onda scende al di sotto della linea orizzontale, il cono si ritrae diminuendo la pressione.

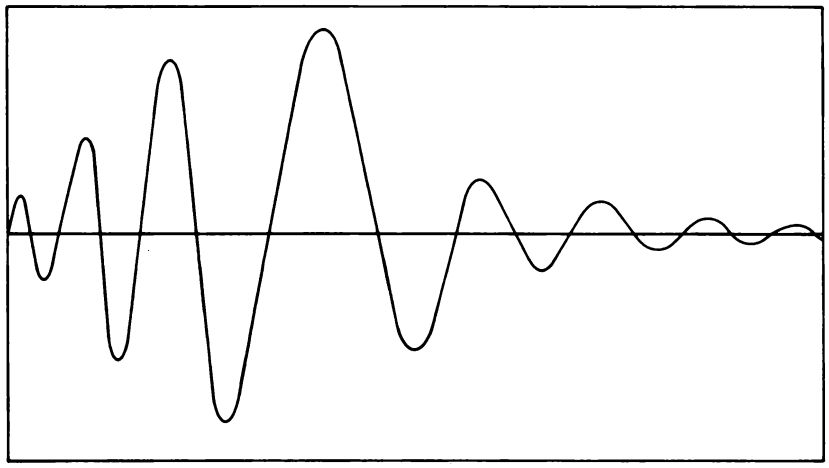


Figura 7-6.

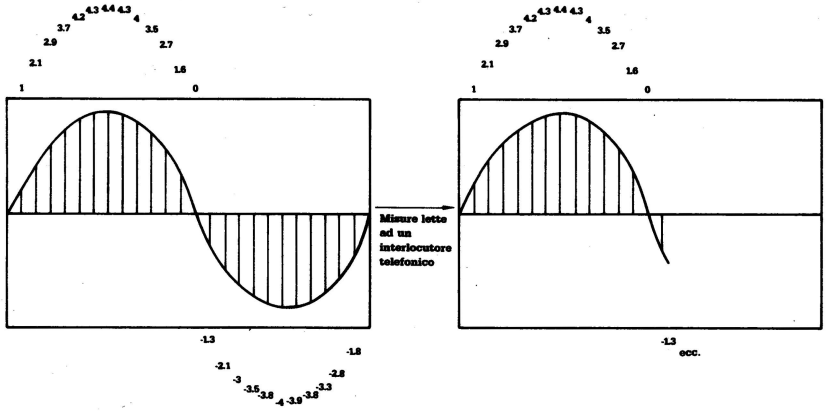
Una tipica forma d'onda.

Leggendo la forma d'onda, è possibile ricostruire tutta la serie di movimenti compiuti dal cono dall'inizio alla fine del suono. Più onde ci sono all'interno della forma, più il cono si muoverà velocemente avanti e indietro. Più in alto (e in basso) si spingono le onde, più il cono è attraversato da vibrazioni. Inviando la forma d'onda di un suono a un altoparlante sotto forma di segnale audio amplificato, è possibile creare il suono in questione.

Per essere immagazzinata nella memoria dell'Amiga, una forma d'onda deve prima essere convertita in forma numerica. Per rendere l'idea di cosa è necessario in questo caso, pensate di dover descrivere l'onda a un'altra persona con cui state parlando al telefono. Sezionate la forma d'onda in piccoli segmenti orizzontali uniformi, come rappresentato nella Figura 7-7 (riportata nella pagina successiva). Se misurate poi l'altezza di ogni segmento, potete trasmettere i dati ricavati al vostro interlocutore telefonico, che può così ricostruire la forma d'onda, purché gli diciate la larghezza dei segmenti che avete usato.

Figura 7-7.

Una forma d'onda sezionata in segmenti regolari. Le misure indicanti l'ampiezza di ogni segmento vengono lette a un interlocutore telefonico che può così ricostruire l'onda senza averla vista.



L'Amiga crea forme d'onda allo stesso modo, memorizzandole poi in una lista ordinata di misure, definita tabella delle forme d'onda. Allo scopo di riprodurre l'onda con precisione, vengono utilizzati segmenti di ampiezza molto ridotta.

Per realizzare forme d'onda complesse e interessanti, l'Amiga ricorre a formule matematiche che, partendo da zero, permettono di giungere a ottimi risultati. Queste formule fissano ogni singolo punto della tabella delle forme d'onda. Una volta completata, la tabella viene inviata a un convertitore digitale-analogico (solitamente definito convertitore D-A) che converte le misurazioni digitali dell'onda in un segnale audio non amplificato; il segnale viene poi amplificato e inviato agli altoparlanti.

L'Amiga ha quattro canali audio interni, ognuno dotato di tavola delle forme d'onda e convertitore D-A. Gli output di questi canali sono raccolti due a due e quindi inviati all'esterno attraverso le porte audio situate sul retro dell'unità centrale dell'Amiga, due dalla porta sinistra e due dalla porta destra. I canali audio riproducono le note rileggendo l'intera tavola delle forme d'onda. È possibile variare la tonalità delle note semplicemente cambiando il modo in cui l'Amiga legge la tavola dell'onda di un canale: più veloce è la lettura, più alto risulterà il suono emesso.

Nonostante abbia solo quattro canali audio, l'Amiga non è limitato a quattro voci musicali; ogni singola tavola delle forme d'onda può immagazzinare un numero infinito di toni, cosicché l'utente può creare un numero illimitato di voci musicali. Pensate ad esempio a un'unica forma d'onda prodotta da un microfono piazzato di fronte a una grande orchestra sinfonica. La forma d'onda in questione contiene informazioni sonore provenienti da più di cento strumenti che suonano contemporaneamente note di altezza diversa. Riproducendo la forma

d'onda con un altoparlante si percepiscono tutte le diverse altezze di tonalità suonate dall'orchestra. Una tavola delle forme d'onda che contenga l'onda prodotta dall'orchestra e che sia memorizzata in uno dei canali audio interni dell'Amiga riprodurrebbe lo stesso numero di tonalità originali.

Nella pratica è molto complicato ottenere più di quattro voci musicali indipendenti con l'Amiga. Creare una forma d'onda su una sola tonalità è facile, ma utilizzare la sintesi matematica per realizzare in un unico canale una tavola di forme d'onda a più toni è molto più difficile. Una volta creato il modello matematico dell'onda, i toni in esso contenuti sono strettamente interdipendenti. Quando il canale audio aumenta o diminuisce la velocità di riproduzione per elevare o abbassare il tono, tutte le tonalità vengono trasformate congiuntamente; non si può dunque parlare di voci musicali indipendenti. Per poter modificare le singole tonalità di un canale audio politonale, l'Amiga deve calcolare una nuova forma d'onda per ogni variazione tonica. L'impresa non è impossibile, ma richiede alcune sottigliezze di programmazione; esistono programmi che riproducono otto o persino sedici voci musicali indipendenti.

Quando usate l'Amiga come sintetizzatore non è necessario che calcoliate le tavole delle forme d'onda che vi interessano (a meno che siate un programmatore esperto). Vi conviene ricorrere alla grande varietà di metodi offerti dai programmi realizzati per l'Amiga. Alcuni programmi visualizzano in appositi menu le diverse possibilità di modifica dei suoni, altri consentono all'utente di disegnare direttamente la forma d'onda desiderata. Indipendentemente dal tipo di controlli offerti, tutti i programmi agiscono sulle formule matematiche che creano la tavola delle forme d'onda all'interno dell'Amiga.

Campionamento dei suoni

L'Amiga e alcuni sintetizzatori digitali possono produrre suoni senza sintetizzarli. Essi registrano semplicemente un suono dall'ambiente circostante e lo immagazzinano in una tavola di forme d'onda come una serie di numeri. Questi suoni registrati sono detti suoni campionati, la misura di ogni segmento dell'onda è detta campione e l'intero processo di realizzazione e immagazzinamento delle misure è definito campionamento.

Per creare un suono campionato con l'Amiga, bisogna possedere un microfono e un campionatore. Il microfono raccoglie un suono e ne invia la forma d'onda al campionatore sotto forma di segnale audio non amplificato. Il campionatore divide l'onda in circa 20.000 segmenti al secondo (numero che varia a seconda del tasso di campionamento), legge l'ampiezza di ciascun segmento e invia ogni singolo campione all'Amiga, nello stesso ordine in cui gli sono giunti. Il flusso di campioni

viene memorizzato dall'Amiga nella tavola apposita e può essere riprodotto attraverso il convertitore D-A di un canale audio, allo stesso modo di una forma d'onda sintetizzata.

I suoni campionati sono generalmente molto ricchi, poiché vengono registrati direttamente da una fonte acustica molto complessa. Il grande svantaggio è che questi suoni richiedono notevoli quantità di RAM, perché la memoria deve immagazzinare ogni singolo campione; nel caso dei suoni sintetizzati, invece, l'Amiga memorizza la forma d'onda con una semplice formula. Se pensate che ogni campione richiede un byte di memoria e che vengono campionati 20.000 segmenti al secondo, se ne ricava che un suono campionato della durata di due secondi richiede uno spazio RAM di 40.000 byte. Il numero sale a dismisura se si utilizzano più suoni campionati contemporaneamente. Al contrario, i suoni sintetizzati richiedono molta meno memoria. Invece di memorizzare l'intera forma d'onda come una serie di numeri singoli, l'Amiga immagazzina la formula matematica che permette di creare il suono in questione. Quando l'utente vuole risentire il suono sintetizzato, l'Amiga richiama la formula e crea la forma d'onda corrispondente.

I suoni campionati sono invariabili se riprodotti normalmente; la forma d'onda viene restituita con la stessa tonalità, volume, timbro e lunghezza del suono originale; non è altro che una registrazione digitalizzata del suono. Fortunatamente l'Amiga può alterare la riproduzione di un suono campionato, permettendo di usarlo a scopi musicali. Aumentando e diminuendo il valore dei campioni, l'Amiga varia il volume dei suoni. Inviando i campioni al convertitore D-A a velocità più o meno elevata, l'Amiga varia invece la frequenza del suono.

Per modificare la durata di un suono campionato, l'Amiga deve continuamente ripeterne una parte. La sezione scelta per la ripetizione deve avere un suono regolare e armonioso, quindi bisogna ascoltare attentamente il suono prodotto da una forma d'onda campionata per decidere quale sia la parte migliore. Se la sezione è troppo vicina all'inizio del suono, potrebbe accadere che nella ripetizione venga continuamente riproposto l'attacco (la corda appena pizzicata di una chitarra, ad esempio), con un effetto poco piacevole. Al contrario, se la sezione prescelta è verso la fine dell'onda, il suono potrebbe ormai essersi affievolito, col risultato che la ripetizione non ha più alcun senso.

Stabilire quale sia la sezione più adatta è un problema di capacità di giudizio. Nei suoni campionati ottenuti con un programma come Deluxe Music la sezione adatta viene automaticamente prescelta e registrata sul dischetto. Per creare suoni campionati propri, l'utente ha bisogno di software da affiancare al campionatore e ai microfoni che utilizza per

catturare il suono originale. Una volta campionati i suoni, il software consente di visualizzarne le forme d'onda sullo schermo e di perfezionarne la parte iniziale e finale fino a ricavare il miglior suono possibile.

SEQUENCER

Il sequencer è uno strumento che controlla e fa suonare uno strumento musicale. Molti conoscono le pianole che utilizzano un rotolo di carta traforata per suonare i tasti del piano. Ogni foro del rotolo corrisponde a un tasto del pianoforte. Un sequencer elettronico funziona sulla base di principi molto simili, poiché imagazzina note con un procedimento elettronico e le utilizza per suonare con un sintetizzatore. I primi sequencer erano piuttosto rudimentali, semplici strumenti elettromeccanici che memorizzavano solo una sequenza di tonalità. Suonavano da 20 a 50 note su un sintetizzatore, tutte dello stesso valore, e poi le ripetevano all'infinito finché non venivano interrotti.

I sequencer moderni sono totalmente differenti. Sono veri e propri computer e non sono più limitati nel numero di note che possono mettere in sequenza. Possono immagazzinare centinaia di migliaia di toni, possono regolarne il valore e altri fattori quali il volume, la strumentazione e il timbro.

I programmi di sequencer sull'Amiga raggiungono livelli qualitativamente molto elevati. È possibile immettere le note utilizzando un tradizionale pentagramma o un sistema di notazione approntato appositamente per il programma. Il software immagazzina le note come una sequenza di eventi collocati nel tempo con precisione. Gli eventi possono essere fattori come l'inizio e la fine di una nota, l'alternanza fra gli strumenti, la variazione di volume e altre caratteristiche musicali. Quando il sequencer ripropone il contenuto immesso, invia tutti questi eventi al sintetizzatore interno dell'Amiga, sotto forma di segnali di controllo. Questi segnali agiscono come un musicista invisibile che inizia e termina le note e si trastulla con le varie manopole del sintetizzatore per modificare la qualità timbrica del suono.

Il vantaggio del sequencer rispetto alla tastiera consiste nel fatto che il sequencer può suonare meglio di qualunque musicista in carne e ossa. Può controllare più voci, suonare musica più veloce e complicata e trasportare un pezzo in un'altra tonalità o in un altro tempo con un unico comando. Lo svantaggio è che i sequencer non sono spontanei. Devono essere programmati e quindi non possono effettuare variazioni sul tema, a seconda dell'atteggiamento del pubblico.

LA SINTESI VOCALE

Se vi ascoltate attentamente mentre parlate, potrete percepire i minimi particolari dell'espressione orale: variazioni di timbro nel passaggio dalle vocali alle consonanti, variazioni di tonalità e volume quando si parla con una certa inflessione, accentando sillabe diverse. Sintetizzando una forma d'onda che contenga tutte queste variazioni sottili e subitanee di timbro, tono e volume, è possibile far parlare un sintetizzatore.

L'Amiga, nel suo software sistema, è dotato di un programma che semplifica notevolmente la sintesi vocale. Questo programma preleva stringhe alfanumeriche da fonti quali la tastiera o la memoria e le trasforma in forme d'onda che, quando vengono convertite in suono dal canale audio, danno origine a parole invece che a musica. Il testo che deve essere verbalizzato viene immesso nel computer foneticamente, utilizzando una speciale compitazione fonetica comprensibile al computer. Si usano cioè stringhe come "LAH5NCHWAE2GIN" per ottenere una parola come "lunchwagon" (vagone ristorante). La fonetica dell'Amiga è trattata nel nono capitolo, insieme agli altri dettagli della sintesi vocale.

La compitazione fonetica (detta spelling) permette di controllare i minimi dettagli della pronuncia e dell'inflessione, anche se ci vuole un po' di tempo per trasformare le parole usate correntemente in termini comprensibili alla macchina. Per facilitare il lavoro, l'Amiga dispone anche di routine software che traducono stringhe di parole normali in stringhe fonetiche, atte a essere sintetizzate e riprodotte.

Con lo stesso programma è possibile alterare anche il carattere della voce dell'Amiga. Si può riprodurre una voce alta o bassa per simulare una voce maschile o femminile, e si possono utilizzare diverse tonalità, nel caso vogliate simulare una voce che canta. L'Amiga può parlare senza alcun accento o con forti inflessioni, e può anche variare il volume e la velocità del discorso.

LA REGISTRAZIONE DI MUSICA ELETTRONICA

Suonare con l'Amiga è molto divertente, ma nessuno vi potrà mai sentire, a meno che non vi piazziate in mezzo alla strada col computer o invitiate un po' di gente nella vostra stanza. La soluzione migliore è registrare le vostre opere musicali su nastro. Se possedete un impianto stereo avrete quasi sicura-

mente un registratore a cassette con il quale potrete ascoltare la musica realizzata con l'Amiga. Potrete duplicare il nastro e spedirlo a parenti lontani e agenti discografici, oppure ascoltarlo con la vostra autoradio mentre guidate l'auto.

La registrazione su nastro offre altri vantaggi: si possono introdurre suoni provenienti da fonti esterne all'Amiga. Ad esempio, immettendo l'accompagnamento di una canzone e riascoltandolo, vi si possono aggiungere le parole utilizzando un microfono, registrando il tutto. È inoltre possibile fondere i due canali dell'Amiga per ottenere un miglior effetto stereofonico, e ricorrere ad alcuni espedienti particolari per registrare l'Amiga che suona con se stesso. Per far questo, serve soltanto un'attrezzatura adeguata.

REGISTRATORI

L'elemento più importante dell'intera attrezzatura è il registratore. Il modello più diffuso è il registratore a cassette, ma potete usare anche un registratore a bobina o un videoregistratore (i videoregistratori ad alta fedeltà offrono prestazioni audio stupefacenti). Il registratore a cassette è senza dubbio più semplice da usare e offre maggiori possibilità di diffondere la vostra musica. Se invece avete intenzione di fare dei montaggi con le vostre registrazioni, un registratore a bobina è senz'altro il mezzo più indicato.

La maggior parte dei registratori è stereo, poiché utilizza due piste distinte. Ogni pista trattiene un segnale audio diverso, che poi riproduce creando l'effetto necessario per la riproduzione sonora stereofonica. Alcuni registratori (in genere quelli a bobina) hanno un numero di piste superiore, che può andare da 4 a 32. Ogni pista può registrare un segnale diverso. Questi registratori (usati nelle sale d'incisione) offrono la possibilità di registrare una voce musicale per ogni pista, in modo da permettere ai tecnici del suono di regolare il volume e il tono di ogni singola pista. Registrando una pista alla volta, un musicista può realizzare completamente da solo un nastro in cui è lui a suonare contemporaneamente tutti gli strumenti.

MIXER

Per inserire diversi segnali in un registratore è necessario possedere un mixer. I mixer consentono di regolare l'intensità di ciascun segnale audio per ottenere le giuste proporzioni fra i suoni. Ad esempio, se avete intenzione di registrare utilizzando un microfono e l'Amiga, potreste notare che il segnale del microfono è più debole di quello dell'Amiga. Il mixer consente di colmare la differenza di intensità dei segnali e inoltre migliora la resa di una riproduzione stereofonica. Il mixer può prelevare i due suoni distinti provenienti dal canale destro e sinistro dell'Amiga e fonderli nei canali del registratore. Un'ap-

posita manopola (detta "pan pot") consente di stabilire la collocazione illusoria dell'origine dei suoni provenienti dall'Amiga.

REGISTRAZIONI SEMPLICI

Se vi basta una semplice registrazione della musica dell'Amiga, non è necessaria un'attrezzatura come quella di cui abbiamo appena parlato. Un registratore a cassette con due input line-level sarà più che sufficiente (un segnale line-level è un segnale audio non amplificato). Le prese audio sul retro dell'Amiga producono un output line-level che può essere collegato direttamente alle entrate di un registratore per mezzo di due cavi audio. Tutto quello che dovete fare è mettere la cassetta nella piastra e iniziare a registrare.

MIDI

Nonostante sia già in grado di produrre incredibili effetti musicali, l'Amiga può essere trasformato in un sistema molto più potente colllegandolo a sintetizzatori esterni per mezzo dell'interfaccia MIDI. MIDI è acronimo di Musical Instrument Digital Interface. La MIDI non costituisce un elemento dell'attrezzatura, è semplicemente un'interfaccia standard che regola il tipo di cavi e di informazioni utilizzati per lo scambio di informazioni musicali fra diversi sintetizzatori o fra un sintetizzatore e un computer.

L'interfaccia MIDI è stata introdotta pochi anni fa per semplificare la connessione di più sintetizzatori. Probabilmente ricorderete ancora, qualche anno fa, quei concerti rock in cui il tastierista balzava avanti e indietro da un sintetizzatore all'altro per suonare un riff con l'unico tra i sintetizzatori a sua disposizione che gli avrebbe fornito il suono necessario per quel pezzo. Quando i sintetizzatori sono collegati per mezzo di cavi MIDI, il musicista può utilizzare la tastiera di uno solo dei sintetizzatori per ottenere il suono di qualunque altro che sia stato collegato in rete. Il musicista può preparare un sintetizzatore centrale da cui ricavare tanto i suoni programmati su quello strumento, quanto i suoni preparati sugli altri. Scegliendo quale sintetizzatore vuole suonare per mezzo della tastiera centrale, il musicista evita di saltare da questa a quella tastiera.

L'interfaccia MIDI è abbastanza versatile da consentire parecchie altre possibilità di collegamento. Nella rete MIDI si può ad esempio includere un computer. La MIDI fornisce le informazioni riguardanti i tasti premuti, l'alternanza degli strumenti e le

altre azioni effettuate da un musicista quando suona, e il computer registra tutti questi eventi: non registra i suoni, bensì la sequenza della performance. In un secondo tempo, il computer può riprodurre la musica inviando gli eventi registrati ai sintetizzatori collegati nella rete MIDI. Il computer utilizzato in questo modo viene definito registratore MIDI. La sua utilità è notevole, soprattutto perché gli eventi MIDI conservati in memoria possono essere modificati facilmente.

LO STANDARD MIDI

Il nome MIDI sta a indicare anche un vero e proprio standard dell'industria musicale. Quasi tutti i fabbricanti di sintetizzatori prevedono una porta MIDI per i loro modelli, in modo da poterli collegare ad altri strumenti predisposti per questa interfaccia. La standardizzazione consente di connettere più elementi MIDI con estrema facilità. Si utilizza un cavo MIDI con due spinotti a 5 poli (spinotti audio standard) posti alle due estremità del cavo, inserendone uno nella porta MIDI di uno strumento e l'altro nella porta MIDI del secondo strumento.

Ci sono tre diversi tipi di porte MIDI: MIDI In, MIDI Out e MIDI Thru. La porta MIDI In riceve il segnale proveniente da un altro apparecchio MIDI. La porta MIDI Out invia il segnale a un altro apparecchio MIDI; questa porta invia generalmente solo segnali provenienti dallo strumento d'origine, ma con alcuni strumenti consente anche di fondere i segnali ricevuti attraverso la porta MIDI In per farli poi passare entrambi dalla porta MIDI Out. La porta MIDI Thru si limita a trasmettere i segnali che entrano dalla porta MIDI In senza modificarli.

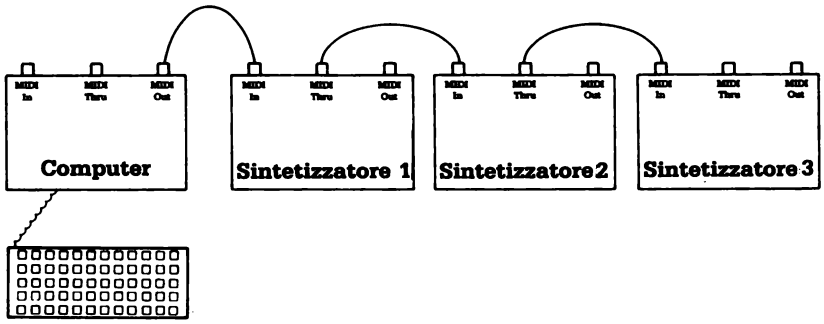
I tre tipi di porte consentono di connettere l'attrezzatura MIDI in configurazioni diverse. Ad esempio, si può connettere un computer a un sintetizzatore collegando la porta MIDI Out del computer alla porta MIDI In del sintetizzatore, e la porta MIDI In del computer alla porta MIDI Out del sintetizzatore. In questo caso il computer è in grado di leggere tutti i segnali MIDI inviatigli dal sintetizzatore e viceversa.

Se possedete un computer e parecchi sintetizzatori, si può utilizzare il primo come centro di controllo dei sintetizzatori stessi. È sufficiente collegare la porta MIDI Out del computer alla porta MIDI In del primo sintetizzatore. Quindi collegate la porta MIDI Thru del primo sintetizzatore alla porta MIDI In del secondo sintetizzatore, e la porta MIDI Thru di quest'ultimo alla porta MIDI In del terzo sintetizzatore. Potete aggiungere tutti i sintetizzatori che volete collegando sempre la porta MIDI In del sintetizzatore aggiunto alla porta MIDI Thru dell'ultimo

sintetizzatore della rete. Con una simile configurazione ogni sintetizzatore è in grado di ricevere soltanto messaggi MIDI inviati dal computer o di ritrasmettere tali messaggi al sintetizzatore che lo segue nella rete di collegamenti. La Figura 7-8 riporta un esempio di tale configurazione.

Figura 7-8.

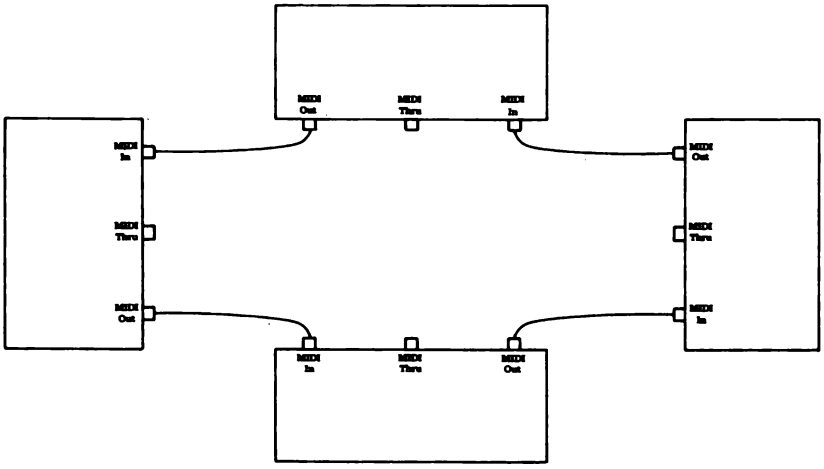
Come collegare un computer per controllare una rete di strumenti MIDI.



Un'altra configurazione che consente lo scambio di informazioni fra diversi strumenti MIDI è il "circuito" MIDI illustrato nella Figura 7-9. Per realizzare un circuito di questo genere, tutti gli apparecchi MIDI devono essere in grado di mixare i segnali MIDI In che ricevono e i segnali che creano, inviandoli poi all'esterno attraverso la porta MIDI Out. La configurazione seguente consente a ciascun apparecchio collegato di scambiare informazioni con tutti gli altri elementi della rete.

Figura 7-9.

Rete di strumenti MIDI che consente l'interscambio di messaggi MIDI fra le varie componenti.



Per usare l'Amiga come macchina MIDI, è necessario aggiungere un adattatore MIDI alla porta seriale sul retro dell'unità centrale. L'adattatore ha tre porte differenti e rende il segnale dell'Amiga compatibile col segnale MIDI standard.

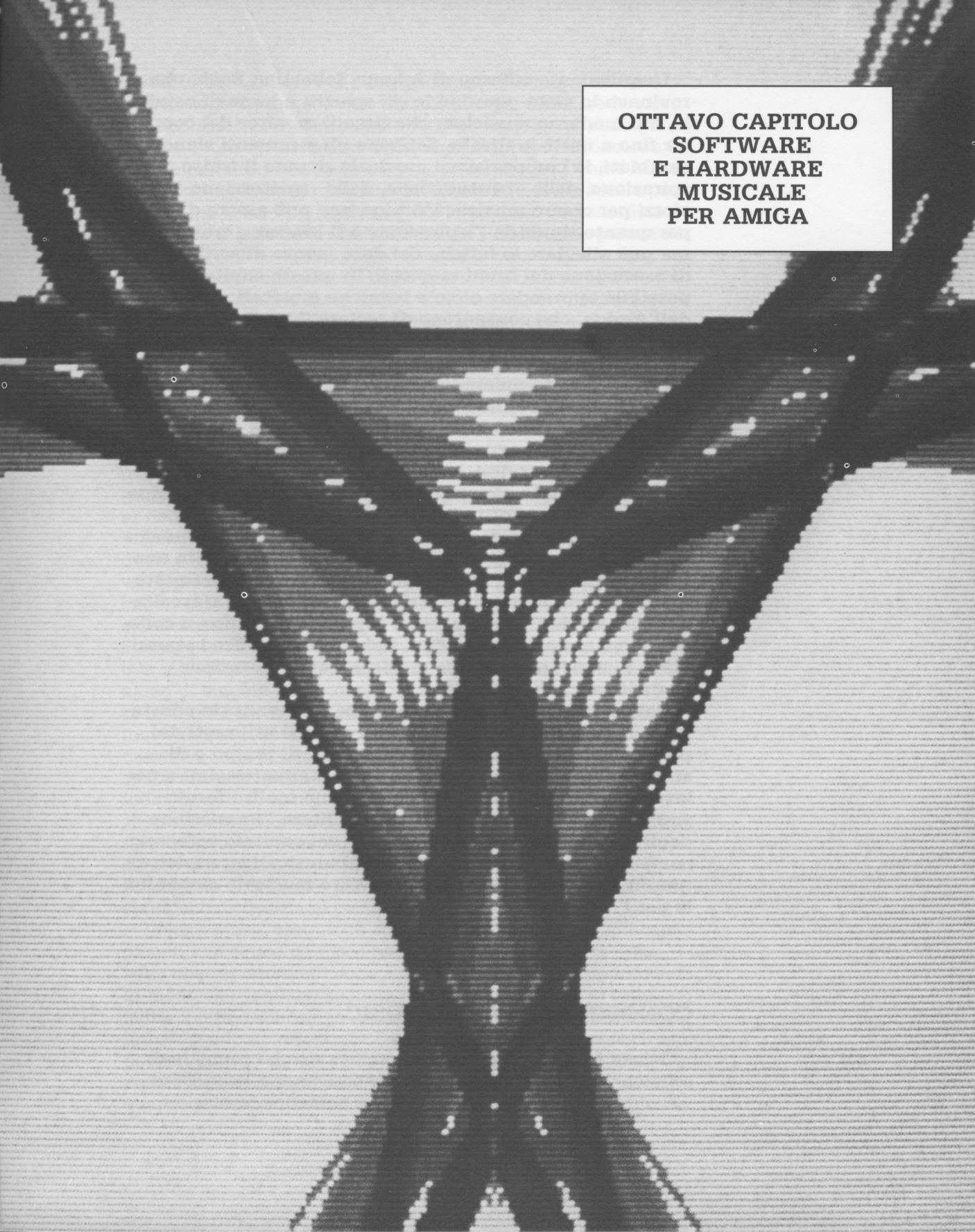
MESSAGGI MIDI

I segnali che passano da uno strumento a un altro per mezzo di cavi MIDI sono simili ai segnali trasmessi da due computer collegati per mezzo di un modem. La comunicazione modem utilizza normalmente il codice ASCII (American Standard Code for Information Interchange, Codice Standard Americano per lo Scambio di Informazioni), che utilizza un codice cifrato per rappresentare le lettere, i numeri, i segni d'interpunzione e altre informazioni utili al computer, come ad esempio l'ordine di andare a capo, l'immissione di linee di programma o di altri caratteri di controllo.

Anche il MIDI utilizza un codice cifrato, che invece di rappresentare numeri e lettere interpreta i diversi eventi che interessano il sintetizzatore. I numeri di codice MIDI sono raggruppati in unità definite messaggi. I messaggi MIDI tipici comunicano messaggi quali "note-on", a indicare che un certo tasto è stato premuto, precisandone la tonalità e la forza con cui è stata esercitata la pressione. Il messaggio "note-off" indica invece che è terminata la pressione di un tasto, precisando naturalmente di quale tasto si tratta. I messaggi MIDI indicano anche i movimenti dei regolatori di tonalità, consentendo di conoscere la variazione apportata al tono modificato.

I messaggi MIDI sono inviati via cavo attraverso 16 canali controllati dal software. Ogni canale contiene i messaggi relativi a un unico apparecchio MIDI (apparecchio e canale corrispondente sono specificati dall'utente); se ne deduce che possono essere collegati in rete fino a 16 strumenti MIDI, ognuno dei quali invia e riceve messaggi attraverso un canale riservato. Ciascun canale MIDI trasmette gli eventi che riguardano il suo sintetizzatore, in modo che ogni canale possa trasmettere un'intera performance: accordi, arpeggi fantasiosi, cambi strumentali e via dicendo. Un computer centrale che controlli i sintetizzatori esterni può utilizzare i canali separati per inviare messaggi e registrare eventi da ogni singolo sintetizzatore.

A questo punto ne sapete abbastanza in materia di origine, percezione e produzione elettronica dei suoni. Abbiamo visto come funziona la sintesi sonora e come l'Amiga può sintetizzare suoni e discorsi propri e ricreare suoni campionati. È stato presentato anche il sistema MIDI, insieme ai principi essenziali che regolano il collegamento di sintetizzatori esterni all'Amiga. Partendo dalle informazioni di base fornite in questo capitolo, sarà ora possibile affrontare i due capitoli successivi dedicati al suono con cognizione di causa. Alla fine sarete in grado di produrre suoni, musica e sintesi vocale.



OTTAVO CAPITOLO SOFTWARE E HARDWARE MUSICALE PER AMIGA

L'equivalente odierno di Johann Sebastian Bach, che si rovinava la vista guardando gli spartiti a lume di candela, è il moderno musicista che fissa il monitor del computer fino a notte inoltrata. Sebbene gli strumenti siano cambiati, la composizione musicale rimane il frutto dell'ispirazione, della strutturazione, della registrazione e degli sforzi per creare musica. L'Amiga non può essere d'aiuto per quanto riguarda l'ispirazione o il processo creativo, ma può alleviare le fatiche del duro lavoro di scrittura e di esecuzione dei brani musicali. In questo capitolo sarà possibile apprendere alcune tecniche musicali utilizzate dall'Amiga, che consentono di restare il meno a lungo possibile davanti al monitor e di trascorrere più tempo ad ascoltare la propria musica.

È stata scritta un'ampia varietà di programmi musicali appositamente studiati per l'Amiga. Alcuni programmi immagazzinano note musicali nella memoria dell'Amiga per poterle riproporre in qualsiasi momento; altri invece consentono di servirsi, come strumento musicale, di una tastiera collegata al computer o della tastiera dell'Amiga stesso. Esistono molti altri programmi con i quali è possibile creare degli strumenti personali per fare musica sintetizzata e modificarli. Il programma illustrato in questo capitolo, Deluxe Music dell'Electronic Arts, permette soprattutto di memorizzare delle note e di farle riascoltare. Utilizzando il sintetizzatore interno dell'Amiga o collegando un sintetizzatore MIDI, Deluxe Music offre la possibilità di scrivere partiture musicali con la notazione tradizionale e di suonarle.

Questo capitolo spiega le tecniche più avanzate adottate da Deluxe Music, presentando i metodi utilizzati dal programma per definire i cambiamenti di tempo e di strumento, per usare le note in sessantaquattresimi e per lavorare con apparecchi MIDI. Verranno inoltre forniti suggerimenti e consigli utili per facilitare l'immissione degli spartiti. Le sezioni successive del capitolo spiegano come registrare le proprie creazioni musicali su nastro e presentano altro software e hardware musicale disponibile per l'Amiga.

CONOSCERE DELUXE MUSIC

Deluxe Music è un programma sofisticato che permette di scrivere, revisionare, suonare e stampare pezzi musicali. Per visualizzare le partiture si serve della notazione musicale

tradizionale – le note sul pentagramma – ma mette anche a disposizione un metodo non tradizionale per facilitare la stesura degli spartiti. È possibile sia scrivere le note sul pentagramma con i loro diversi valori utilizzando il mouse, sia inserire i toni selezionandoli da una tastiera musicale che Deluxe Music visualizza nella parte inferiore del monitor, metodo ideale per chi ha una certa dimestichezza con la tastiera. Se possedete una tastiera MIDI collegabile all'Amiga è sufficiente premere i tasti e le relative note saranno inserite e memorizzate.

Il sistema di immissione degli spartiti di Deluxe Music offre possibilità alquanto interessanti. Possono essere visualizzati sullo schermo fino a otto pentagrammi alla volta e si può scegliere fra la chiave di violino, di basso, di contralto e di tenore. È inoltre possibile inserire note e pause comprese fra i valori di intero e di trentaduesimo, aggiungere un punto per alterare il valore di qualsiasi nota e usare queste ultime per creare terzine e quintine. Si possono inoltre introdurre legature di valore, inserire fasi in crescendo e diminuendo, immettere marcature dinamiche che variano da *ppp* (molto dolce) a *fff* (molto forte) e cambiare le indicazioni del tempo e la segnatura delle chiavi all'interno dello spartito.

Deluxe Music offre notevoli opzioni creative per mezzo delle quali si può rielaborare la musica nello stesso modo in cui un word processor elabora i testi. Deluxe Music ha caratterististiche in grado di rendere la partitura più piacevole alla vista sia sullo schermo che sulla copia stampata. È inoltre possibile stabilire la direzione dei gambi delle note e unire gruppi di ottavi, sedicesimi e trentaduesimi per maggior chiarezza. L'unione di singole note, come mostra la Figura 8-1 (riportata a pagina 217), facilita la lettura al musicista. Note, pause e rigo musicale possono essere cambiati di posizione per ottenere il massimo della chiarezza ed è anche possibile aggiungere allo spartito il testo della canzone, la segnatura del tempo o le indicazioni utili al musicista. Deluxe Music possiede una speciale fonte musicale incorporata che racchiude in sé simboli musicali quali le legature di valore, i suoni vibrati e altri caratteri utili in una partitura. La partitura, quando ha raggiunto la forma desiderata, può essere riprodotta mediante la stampante collegata al computer.

Deluxe Music esegue gli spartiti utilizzando strumenti campionati che carica da disco nella memoria dell'Amiga. Una volta che gli strumenti sono stati memorizzati, il programma fa in modo che vengano suonati, servendosi dei quattro canali audio interni dell'Amiga. Lo spartito può essere orchestrato introducendo in vari punti delle variazioni di strumento. L'uso dei soli quattro canali audio pone dei limiti a Deluxe Music, poiché è possibile suonare contemporaneamente non più di quattro sole note. Tuttavia non è esclusa la possibilità di

inserire ulteriori voci, mediante un collegamento con un sintetizzatore MIDI. Deluxe Music potrà in questo modo usare gli strumenti del sintetizzatore esterno in aggiunta a quelli dell'Amiga. Si tratta di un'opzione molto funzionale se consideriamo la possibilità, offerta da Deluxe Music, di inserire simultaneamente un gran numero di note disposte su addirittura otto righi musicali.

COME AMPLIARE LE CAPACITÀ DI DELUXE MUSIC

È abbastanza scontato e comune servirsi di Deluxe Music per introdurre lo spartito copiandolo direttamente da un foglio. Le note vengono in questo modo copiate una per una allo scopo di consentire all'Amiga di riprodurre l'intero pezzo. Inserire le sole note e le pause riportate sul foglio di musica è un'operazione abbastanza semplice. Sfortunatamente, la musica memorizzata in tale modo ha un suono meccanico e poco incisivo. Le note e le pause costituiscono infatti solo la struttura della musica; la sua espressività è frutto esclusivo della versatilità e spontaneità che il musicista dimostra nell'esecuzione dei brani.

Deluxe Music offre diverse opzioni che contribuiscono a sottolineare lo stile della partitura anche nella riproduzione del pezzo musicale; è possibile utilizzare crescendo e diminuendo, nonché una segnatura dinamica per aumentare e diminuire il volume della musica, conferendole una certa espressività. Inoltre si possono legare gruppi di note in modo che non vi siano interruzioni di suono fra l'una e l'altra, oppure si possono creare note in staccato, in modo da udirle singolarmente. Queste caratteristiche si fondano sull'individualità e indipendenza di ogni singola nota: l'azione può iniziare da qualsiasi nota della partitura e concludersi ugualmente su una nota a piacere.

Deluxe Music dispone di altre peculiarità che permettono di dare un certo brio alle vostre opere musicali. Ad esempio, è possibile iniziare a produrre un suono utilizzando uno strumento e, in seguito, cambiare strumento per ottenere un timbro diverso: cambi di timbro così improvvisi da far rizzare le orecchie all'esperto più consumato. È anche possibile modificare a metà partitura la velocità di riproduzione; anche l'accelerazione o il rallentamento repentino dell'esecuzione possono avere effetti inattesi sull'ascoltatore. Sebbene queste due opzioni siano particolarmente utili, Deluxe Music non consente di applicarle a partire da una nota qualsiasi dello spartito. Possono infatti essere utilizzate soltanto all'inizio o alla fine di una battuta, a metà della quale non è assolutamente possibile cambiare tempo o strumento.

La musica adopera generalmente le battute o misure solo come riferimento temporale; pertanto le barre di divisione fra

una battuta e l'altra non sempre cadono nel punto preciso in cui si desidererebbe cambiare uno strumento o il tempo. In effetti, se si intende ricorrere all'accelerando (in cui il tempo aumenta costantemente a ogni gruppo di note), o al ritardando (in cui il tempo diminuisce analogamente), la capacità di Deluxe Music di cambiare tempo a ogni misura non produrrà l'effetto che ci si aspetta quando si usano le battute ordinarie.

Fortunatamente Deluxe Music consente di creare battute di quasi tutte le dimensioni, in modo da aggirare l'ostacolo costituito dalle limitazioni imposte alle battute per ciò che riguarda i cambiamenti di tempo e di strumento. Utilizzando alcuni dei comandi del menu Measures, come Insert Measure, Split Measure e Join Measure, e stabilendo in seguito una diversa indicazione del tempo per ciascuna battuta, è possibile regolare le battute di uno spartito in modo che cadano esattamente dove si desidera introdurre le variazioni. La prossima sezione mostrerà come sfruttare al meglio questa peculiarità.

I cambiamenti di tempo: accelerando e ritardando

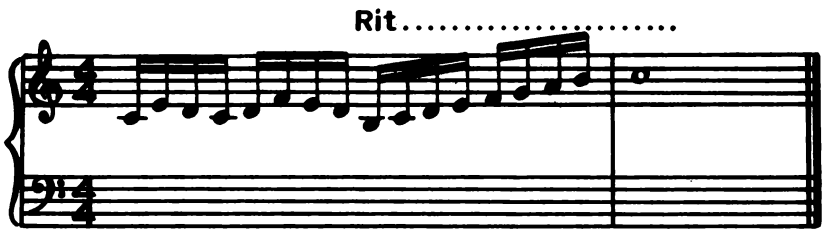


Figura 8-1.

Un esempio di ritardando applicato a due misure musicali.

La Figura 8-1 riporta le ultime due battute di un componimento musicale. Per ottenere un buon suono di chiusura viene utilizzato un ritardando (indicato con "rit" seguito da una linea punteggiata) che ritarda la nota semibreve finale (o intera) rallentando gli ultimi otto sedicesimi. Se intendete provare questo breve passaggio, seguite le seguenti istruzioni:

1. Usate il comando **New Score** del menu **File** per creare un nuovo spartito.

2. Usate il comando **Set Time Signature** del menu Measure per assicurarvi che la segnatura del tempo dello spartito sia regolata sui 4/4.

3. Immettete le note come sono nella Figura 8-1: sedici semicrome nella prima misura e una semibreve nella seconda.

4. Selezionate **Score Setup** dal menu **Window** e assicuratevi che il regolatore **Beats per Min** sia regolato su 90.

5. Provate a eseguire la partitura. La riproduzione dovrebbe consistere in 90 battute al minuto senza alcun ritardando.

6. Usate la freccia di composizione per visualizzare il cursore necessario all'inserimento di note (una linea verticale a intermittenza) fra l'ottava e la nona nota della prima battuta.

7. Utilizzate il comando **Split Measure** del menu **Measures** per inserire una nuova barra nella posizione del cursore, in modo da dividere la prima battuta in due battute da otto note ciascuna.

8. Utilizzando la freccia e il comando **Split Measure**, dividete la seconda delle due battute appena create in quattro battute di due sedicesimi ciascuna.

A questo punto avrete ottenuto uno spartito di sei battute: la prima di otto note, le battute dalla seconda alla quinta di due note e la sesta di una nota semibreve. Per vedere contemporaneamente tutte e sei le misure, utilizzate la freccia per spostare verso sinistra la barra divisoria destra di ogni misura, finché tutto lo spartito sarà visibile sullo schermo.

Riproducendo lo spartito, noterete che vi sono pause piuttosto lunghe nei punti in cui sono state inserite le nuove barre. Ciò è dovuto al fatto che tutte le battute misurano 4/4 e sono riempite solo in minima parte. Per ripristinare l'andamento ritmico sconvolto è necessario cambiare il tempo delle nuove battute:

1. Cambiate il tempo della prima battuta in 2/4, sempre servendovi della freccia e utilizzando il comando **Set Time Signature...** del menu **Measures** per definire il nuovo tempo di 2/4.
   Deluxe Music cambia automaticamente l'indicazione della prima battuta e di tutte quelle successive. Pertanto, quando la prima misura è stata cambiata in 2/4, tutte le battute seguenti subiscono la stessa variazione.
   Supponiamo però che vogliate utilizzare il tempo di 2/4 solo per la prima battuta: è chiaro che dovrete cambiare il tempo per le misure successive.

2. Cambiate il tempo della seconda, terza, quarta e quinta battuta in 1/8, inserendo l'indicazione prima della

seconda battuta, utilizzando ancora una volta la freccia e il comando **Set Time Signature...**.

3. Ripetete la stessa operazione per ristabilire il tempo di 4/4 prima della sesta e ultima battuta. A questo punto ogni segnatura di tempo della misura è regolata in modo che i contenuti della misura la riempiano completamente.

Riproponendo il tutto, ci si accorgerà che il risultato è esattamente lo stesso della prima riproduzione. Vi sembrerà di aver fatto tanto lavoro per niente, ma ricordatevi che ora, frammiste alle note, vi sono numerose sbarre che vi consentono di aggiungere un ritardando:

1. Fate eseguire lo spartito con tempi differenti variando la regolazione del cursore **Beats per Min** nella finestra **Score Setup**, al fine di trovare un tempo lento che vi sembri adatto per il ritardando nell'ultima battuta. In questo caso si adattano perfettamente 90 battute al minuto come tempo iniziale che diminuisce poi fino a 56 prima della semibreve finale.

2. Stabilite come tempo della prima misura 90 battute al minuto. Per farlo, selezionate la prima battuta, assicuratevi che il cursore **Beats per Min** nella finestra **Score Setup** sia regolato su 90 e infine utilizzate il comando **Set Tempo** del menu **Measures**. L'indicazione del tempo apparirà all'inizio della misura.

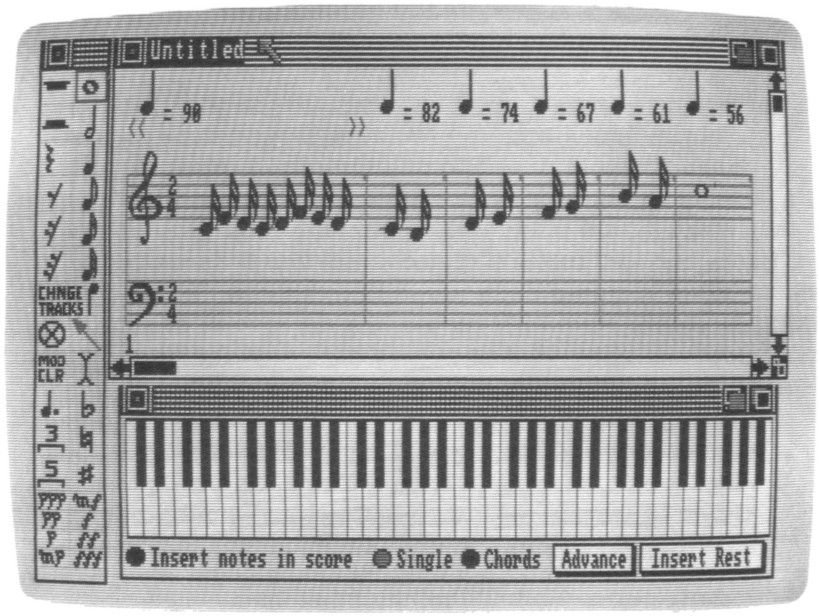
3. Fissate come tempo per l'ultima misura 56 battute al minuto. Per farlo, selezionate l'ultima misura, regolate su 56 il cursore nella finestra **Score Setup** e infine immettete il comando **Set Tempo**.

4. Per creare un ritardando, trasformate i tempi delle quattro misure intermedie in incrementi variabili da 90 a 56. Valori quali 82, 74, 67 e 61 si adattano bene. I tempi dello spartito finale dovrebbero quindi essere: 90, 82, 74, 67, 61 e 56.

La versione finale deve risultare uguale allo spartito della Figura 8-2 nella pagina successiva. Ascoltando nuovamente il pezzo potrete finalmente sentire il ritardando: con l'avvicinarsi della fine della partitura il ritmo rallenta progressivamente.

Figura 8-2.

La divisione delle misure e il tempo progressivamente rallentato permettono di creare un ritardando con Deluxe Music.



Se si applica questo metodo con due o più righi del pentagramma, si noterà che ogniqualvolta si divide una misura su un rigo, Deluxe Music divide allo stesso modo tutte le misure corrispondenti degli altri righi. Dividendo ad esempio la terza misura del primo pentagramma, si otterrà automaticamente la divisione delle terze misure di ogni rigo. Per ovviare a questo inconveniente bisogna dividere le note dei righi non interessati dal ritardando in modo che si adattino alle battute del rigo che contiene il ritardando stesso. Supponendo che, nell'ultimo esempio, ci sia anche un secondo rigo con una semibreve (4/4) nella prima misura, questa dovrà essere divisa in una minima (2/4) e quattro crome (1/8 ciascuna) per fare in modo che si adatti alle nuove battute; dopodiché le cinque note così ottenute vanno unite con una legatura di valore per farle apparire come una sola nota semibreve.

I cambi di strumento

Se si utilizza Deluxe Music con le sole voci interne dell'Amiga, si possono suonare contemporaneamente solo quattro strumenti diversi. Sebbene sia possibile ottenere comunque un considerevole cromatismo anche con quattro strumenti, non

mancheranno le volte in cui ne saranno richiesti di più. Per creare l'illusione di più di quattro strumenti suonati contemporaneamente, è sufficiente inserire nello spartito un numero maggiore di voci alternandole velocemente l'una all'altra.

Pensate ad esempio a una sezione ritmica in un gruppo musicale i cui strumenti siano il basso, il tamburo, il cembalo e la grancassa. Quando suonano assieme, il basso suona in battere mentre gli strumenti a percussione si alternano per riempire le battute in levare. Inserendo in Deluxe Music uno spartito di questo genere, in cui gli strumenti non suonano contemporaneamente, l'Amiga utilizzerà solo una delle sue quattro voci interne. Si potranno effettivamente avere quattro strumenti al "prezzo" di una voce. Nella Figura 8-3 è illustrata una sequenza ritmica che include cinque strumenti differenti.

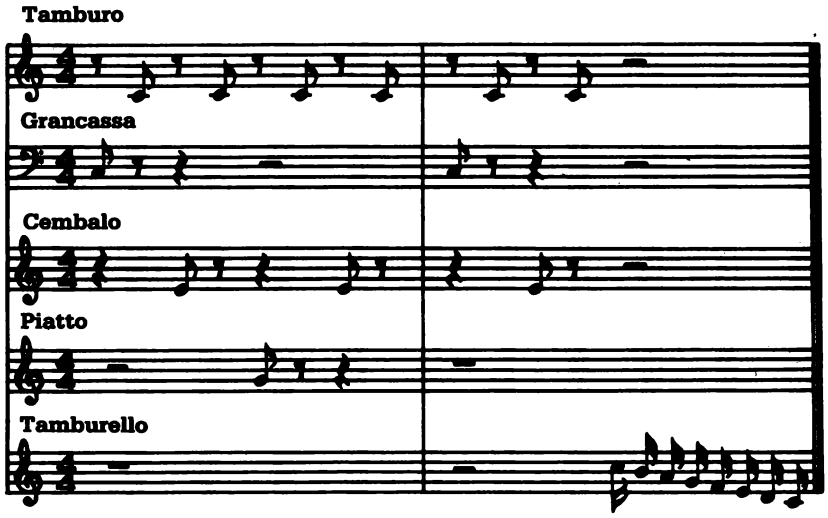


Figura 8-3.

Una sequenza ritmica prodotta da un'alternanza di strumenti.

Per provare le due misure della Figura 8-3 in uno spartito di Deluxe Music, seguite queste istruzioni:

1. Create un nuovo spartito. Aprite la finestra **Score Setup** e aggiungete tre pentagrammi allo spartito. Otterrete così cinque righi musicali per i cinque differenti strumenti. Fissate un tempo di 4/4 in DO maggiore.

2. Caricate in memoria dal dischetto gli strumenti per lo spartito. Ve ne servono cinque: una grancassa, un tamburo, un cembalo, un timpano e un tamburello (bass drum, snare drum, cymbal, clave e tom drum). Se non avete questi cinque strumenti, potete usarne altri a vostro piacimento che abbiano suoni simili. Per caricarli, selezionate il comando **Load Instrument...** dal menu **Sounds** per cinque volte, scegliendo ogni volta uno strumento diverso, in modo che tutti e cinque appaiano alla fine nella parte superiore del menu.

3. Per assegnare il tamburo al primo pentagramma, selezionate la prima misura del rigo musicale, scegliete il tamburo dal menu **Sounds** e, in seguito, **Set Instrument** dal menu **Measures**. Il nome dello strumento appare appena sopra la parte iniziale della prima misura.

4. Usate lo stesso metodo per assegnare la grancassa alla prima misura del secondo pentagramma, il cembalo alla prima misura del terzo rigo, il timpano alla prima misura del quarto e infine il tamburello alla prima misura del quinto.

5. Per inserire le note del tamburo, selezionate la prima misura del rispettivo pentagramma e immettete tutte le note del tamburo, introducendo le pause nei punti in cui vi saranno le note degli altri strumenti.

6. Utilizzate la stessa tecnica per inserire le note di tutti gli strumenti nei rispettivi pentagrammi.

Riascoltando lo spartito, sentirete un'intera sezione ritmica che utilizza soltanto una voce dell'Amiga per tutte le note. È possibile aggiungere più voci alla musica, qualora lo si desideri, servendosi di altri tre pentagrammi e inserendo in essi una parte che impiega uno strumento differente in ogni rigo musicale. Ad esempio, è possibile avere una chitarra solista nel sesto pentagramma, un sassofono nel settimo e un pianoforte nell'ottavo.

Mentre inserite le note per queste nuove voci (o per qualsiasi altro spartito a quattro voci), badate a come utilizzate gli accordi. Non è escluso che esauriate rapidamente tutte le voci dell'Amiga, con l'effetto di tralasciare alcune note. Se, ad esempio, inserite un accordo di tre note nel rigo del pianoforte nello stesso momento in cui stanno suonando anche le altre tre voci, otterrete in totale sei note. Deluxe Music può però suonarne solo quattro e quindi due rimarranno escluse.

Per assicurarsi che gli strumenti importanti abbiano la precedenza nel caso in cui ci siano troppe note contemporaneamente, si può utilizzare una segnatura più marcata per il pentagramma che si vuole avere in risalto. Deluxe Music suonerà sempre per prime le note di un pentagramma regolato con una maggiore dinamicità. Se ad esempio ci sono cinque pentagrammi differenti suonati contemporaneamente e regolati su *pp*, *mp*, *mf*, *f* e *ff*, Deluxe Music suonerà solo sui quattro pentagrammi più marcati. Regolando un pentagramma su un grado leggermente più alto degli altri, ad esempio su *f* se gli altri sono impostati su *mf*, è possibile assicurarsi che quel pentagramma venga sempre sentito distintamente, pur non essendo particolarmente alto rispetto agli altri.

Come usare i sessantaquattresimi

Deluxe Music offre una vasta gamma di lunghezze per le note: esistono sei note di base a cui si possono associare punti e segni di terzine e quintine, fino a raggiungere un totale di 36 note diverse, molte delle quali sono decisamente inconsuete. Con un tale numero di note a disposizione, è ancora possibile andare al di là delle capacità di Deluxe Music: vi potrebbe accadere di copiare uno spartito, giungere alla misura 145 e bloccarvi esterrefatti di fronte a una sfilza di semibiscrome. Cosa fare a quel punto?

La soluzione è semplice e si basa sulla considerazione secondo la quale le lunghezze delle note sono puramente relative, non hanno durata fissa. Ad esempio, inserendo una partitura di semiminime e crome e riascoltandola con un tempo di 60 battute al minuto, le semiminime (1/4) dureranno esattamente un secondo ciascuna, mentre le crome dureranno mezzo secondo. Portando le battute a 120 al secondo, ogni quarto durerà mezzo secondo e ogni croma un quarto di secondo. La durata della nota è molto mutevole, e dipende dal tempo con cui si suona la musica. Tuttavia, le lunghezze relative delle note sono fisse: una semiminima vale sempre due crome e una minima vale sempre due quarti, e così via.

Quando si scrive un pezzo musicale che comprende note di diverse lunghezze, in realtà non si usa una serie di valori temporali, bensì una serie di lunghezze relative: una nota è il doppio dell'altra, un'altra è il triplo e così via per tutto lo spartito. Non importa veramente quali note vengano utilizzate, ciò che conta è la lunghezza della singola nota in relazione alle altre.

Quando vorrete copiare una serie di sessantaquattresimi, non troverete nella palette di Deluxe Music note di valore pari alla metà delle biscrome (1/32). Tuttavia è possibile ricreare le proporzioni raddoppiando il valore di tutte le note della

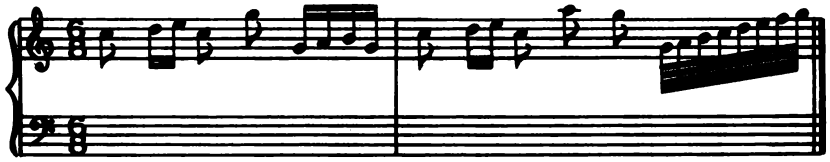
partitura e inserendo i sessantaquattresimi come trentaduesimi. Cimentatevi nell'esempio riportato qui di seguito:

1. Create un nuovo spartito e fissate un tempo di 6/8 in DO maggiore.

2. Inserite le note riportate nella Figura 8-4 fino a raggiungere le semibiscrome alla fine della seconda misura.

Figura 8-4.

Spartito con un finale in sessantaquattresimi.



3. Per adattare le semibiscrome dovete raddoppiare la lunghezza di tutte le altre note dello spartito. Scegliete **Select All** dal menu **Edit** per selezionare l'intero spartito. Selezionate **Double Time** dal menu **Notes**, in modo da raddoppiare il valore di tutte le note (gli ottavi diventano quarti, ecc.). Ma a questo punto le note della seconda metà di ogni misura risulteranno adombrate, perchè il comando **Double Time** non ha alcun effetto sulla durata delle misure. Deluxe Music offusca le ultime note di ogni misura per indicare che non corrispondono più alla segnatura del tempo fissata per quella misura.

4. Per adattare la misura alla nuova lunghezza delle note, dovrete raddoppiare la segnatura del tempo. Selezionate la prima misura dello spartito e regolate la segnatura del tempo sui 6/4 utilizzando il comando **Set Time Signature...** dal menu **Measures**.

5. Inserite le semibiscrome come biscrome alla fine della seconda misura. Eseguendo ora lo spartito, le note avranno sempre la stessa lunghezza relativa l'una rispetto all'altra, ma i loro valori individuali sono stati raddoppiati e la velocità del tempo di esecuzione risulta perciò dimezzata rispetto all'originale.

6. Raddoppiate il tempo di esecuzione selezionando **Score Setup** dal menu **Window** e regolando il cursore **Beats per Min** su un numero di battute doppio rispetto a quello attualmente fissato. Conclusa l'operazione potete applicare il nuovo tempo alla prima misura dello spartito selezionandola e scegliendo **Set Tempo** dal menu **Measures**.

Riascoltando la partitura così modificata, il risultato sarà identico allo spartito originale che avete copiato dal foglio di musica. La differenza è che avete introdotto uno spartito che usa i trentaduesimi per riprodurre il ritmo scritto originariamente in semibiscrome.

Se proseguite l'immissione di note nello spartito in cui avete appena raddoppiato il tempo, dovete ricordarvi di raddoppiare il valore di tutte le note e delle pause. Per fare prima potreste inserire le note e le pause così come sono scritte sul foglio per poi selezionarle e raddoppiarle col comando **Double Time**.

Tutte le volte che adoperate questo metodo, assicuratevi di raddoppiare la segnatura del tempo di ogni battuta. Nel caso di cambiamenti di indicazione del tempo, il raddoppiamento del tempo della prima misura non comporta automaticamente l'adattamento delle altre misure, i cui valori devono essere modificati individualmente. Per raddoppiare il tempo è sufficiente dimezzare il numero che sta al denominatore della frazione. Ad esempio, un tempo di 4/4 raddoppiato diventa un 2/4, un tempo di 3/8 diventa un 3/4. Se non è possibile dimezzare il denominatore non fate altro che raddoppiare il numeratore; ad esempio, 3/1 diventa 6/1.

DELUXE MUSIC E GLI APPARECCHI MIDI

Deluxe Music dispone di una caratteristica molto funzionale: la possibilità di essere utilizzato con un sintetizzatore MIDI in modo da accrescere il numero di strumenti che possono essere suonati contemporaneamente nelle partiture. Per creare una partitura e suonarla in seguito con un sintetizzatore MIDI, viene utilizzato il comando **Set Instrument** nel menu **Measures**, con un'operazione simile a quella che consente di definire uno strumento per l'Amiga.

Per utilizzare uno o più sintetizzatori MIDI con Deluxe Music, è necessario collegarli prima all'Amiga con un'interfaccia MIDI. Tale collegamento può essere effettuato unendo, via cavo MIDI, la porta MIDI Out del primo (o unico) sintetizzatore alla porta MIDI In dell'interfaccia; poi, servendovi di un altro cavo, collegate la porta MIDI Out dell'interfaccia alla porta MIDI In del sintetizzatore. Per aggiungere altri sintetizzatori, collegate la porta MIDI Thru del primo sintetizzatore alla porta MIDI In del secondo, la porta MIDI Thru del secondo alla porta MIDI In del terzo, e così via, finché tutti i sintetizzatori saranno collegati. Esempi di questi collegamenti, insieme a una spiegazione più dettagliata dello standard MIDI, sono riportati alla fine del settimo capitolo.

Ogni sintetizzatore deve essere predisposto in maniera tale da ricevere i messaggi MIDI sul proprio canale MIDI (se usate un solo sintetizzatore, il problema non si pone). Per sapere come regolare il canale MIDI d'arrivo, è necessario leggere i manuali di istruzioni dei sintetizzatori. Se non si riesce a regolare il canale MIDI d'arrivo del sintetizzatore, è molto probabile che i messaggi vengano ricevuti sul canale MIDI 1.

Una volta effettuati i collegamenti potete finalmente accendere i sintetizzatori e usarli con Deluxe Music. Se intendete suonare parte del pezzo musicale immesso con il sintetizzatore, dovete prima predisporre il MIDI utilizzando i comandi del menu **Sounds**.

1. Selezionate il comando **MIDI Active** per caricare in memoria dal dischetto di Deluxe Music il driver MIDI.

2. Utilizzate il comando **MIDI Channel...** per selezionare ciascuno dei 16 canali MIDI che volete usare. Poiché tutti i sintetizzatori collegati sono regolati su canali differenti, potete scegliere qualsiasi sintetizzatore semplicemente selezionandone il canale.

3. Selezionate il comando **MIDI Setup...** per aprire un submenu (requester) in cui potete scegliere un'opzione prefissata. Tali opzioni corrispondono ai diversi suoni strumentali che il sintetizzatore può emettere. Probabilmente vi è capitato di suonare una tastiera elettronica in cui era sufficiente premere dei tasti per ottenere suoni diversi: "Tromba", "Flauto" e altri. Ebbene, questi sono tutti esempi di suoni preselezionati. I sintetizzatori MIDI, in genere, numerano questi suoni in modo da consentire la selezione con un comando che usa un semplice numero. Ancora una volta, per scoprire di quali numeri si tratta, dovrete leggere il manuale del sintetizzatore; trovati i numeri, potete scegliere uno dei numeri prefissati nel requester servendovi del cursore **MIDI Preset Number** per ottenere dal sintetizzatore il suono desiderato. Selezionate infine **OK** per chiudere il requester.

Ora che è stato creato il collegamento MIDI necessario a ottenere dal sintetizzatore il suono desiderato, inserite quanto avete prestabilito nella partitura, nello stesso modo in cui inserireste uno strumento.

1. Selezionate la misura nella quale volete che il sintetizzatore dell'Amiga cominci a suonare.

2. Selezionate il comando **Set Instrument** dal menu **Measures**. La regolazione MIDI da voi predisposta apparirà sopra la misura.

È possibile cambiare il sintetizzatore e i suoni prefissati in qualsiasi punto dello spartito, variando la regolazione MIDI nel menu **Sounds** e, in seguito, utilizzando il comando **Set Instrument** nello stesso modo in cui lo usate per cambiare uno strumento all'interno della partitura.

CONSIGLI UTILI

I consigli che seguono facilitano notevolmente l'immissione di partiture in Deluxe Music.

Segnatura delle partiture stampate

Se vi capita spesso di copiare spartiti stampati, fate in modo di avere accanto a voi parecchie matite colorate. Prima di iniziare a usare Deluxe Music, esaminate la partitura e numerate tutte le misure. Riesaminate la spartito una seconda volta, segnando ora le note che intendete suonare con strumenti diversi. Servendosi di una matita colorata per ogni strumento, sarà più semplice individuare i passaggi in cui i vari strumenti suonano, i punti in cui le note si fondono e gli attacchi di ciascuno strumento.

Mentre si introduce la partitura in Deluxe Music è possibile far corrispondere i numeri delle misure della partitura computerizzata a quelli della partitura su carta da musica, per essere sicuri di inserire correttamente le note. Naturalmente, se le misure vengono sezionate per introdurre un ritardando o altri effetti, è probabile che i numeri delle misure non risultino perfettamente identici. Per evitare questo inconveniente potete segnare le divisioni delle battute anche sulla carta pentagrammata prima di iniziare a copiare. Per immettere le note nel rigo giusto è consigliabile assegnare un colore a ogni pentagramma e quindi inserire in ciascuno di essi solo le note che avete segnato con lo stesso colore.

Uso di abbreviazioni dei comandi

La palette delle note è così in vista sullo schermo che spesso ci si dimentica che non è obbligatorio farne uso. Deluxe Music consente di ricorrere ad alcune abbreviazioni per la tastiera che vi evitano di selezionare la lunghezza della note da introdurre nello spartito. Per saperne di più, consultate il manuale e usate queste "scorciatoie". Quando un'ispirazione vi balena nella mente e siete quasi pronti per mettere la melodia sul pentagramma, è di gran lunga più semplice premere un tasto funzione che non scegliere il valore di una nota muovendo il puntatore sullo schermo per cercare la nota, la disposizione sul pentagramma e l'ispirazione persa nel frattempo.

REGISTRAZIONE SU CASSETTA

Per effettuare una registrazione su cassetta della propria musica, è sufficiente collegare l'Amiga al registratore, accenderlo e far eseguire la musica al computer. Le precisazioni necessarie per registrare con Deluxe Music sono riportate nelle righe che seguono.

Il cavo più semplice per il collegamento fra Amiga e registratore è un cavo stereo con due spine phono a ogni estremità che trasmetta due segnali distinti dall'Amiga al registratore. Per connettere il cavo, inserite le spine di una delle estremità nelle due prese audio jack poste sul retro dell'unità centrale. Fate attenzione all'inserimento delle spine nelle relative prese; osservando l'Amiga dal retro, la presa alla vostra destra è il canale stereo sinistro, mentre la presa a sinistra è il canale destro (sembrerebbe il contrario, ma tenete presente il fatto che, guardando l'Amiga frontalmente, il canale destro è veramente a destra e quello sinistro a sinistra). Sul retro del registratore ci sono due prese probabilmente contrassegnate da REC IN o LINE IN; inoltre la presa del canale di destra è indicata da R e quella del canale di sinistra da L. Inserite le spine nelle prese in modo da collegare i canali destro e sinistro dell'Amiga ai canali rispettivamente destro e sinistro del registratore.

Per registrare la musica, attenetevi alle istruzioni che seguono.

1. Caricate Deluxe Music e la partitura che intendete registrare.

2. Per regolare il volume di registrazione, trovate il punto della partitura in cui la musica giunge al volume più elevato. Selezionate questa parte in modo da poterla utilizzare con **Repeat Play** e ascoltatela più volte. Mentre la ascoltate, regolate il volume del registratore sul livello che più si avvicina, senza toccarlo, al limite in cui inizia la distorsione. Terminata l'operazione, fermate l'esecuzione dello spartito.

3. Quando siete pronti per la registrazione, assicuratevi che il nastro della cassetta sia avvolto in modo da iniziare nel punto desiderato. Se fate partire la cassetta dall'inizio, fate avanzare il nastro finché la parte bianca passa oltre le testine, per evitare che la prima parte della musica che volete registrare non venga incisa sul nastro.

4. Per iniziare la registrazione premete il tasto (o i tasti) Record del registratore, dopodiché usate il comando **Play Song** per suonare la partitura. Terminata l'esecuzione, fermate la registrazione premendo il tasto Stop del registratore.

EFFETTI SPECIALI CON IL TASCAM MINISTUDIO PORTA ONE

Quando si utilizza un registratore collegato direttamente all'Amiga, viene registrato soltanto ciò che l'Amiga trasmette attraverso le sue porte audio: le voci A e D provenienti dal canale sinistro e le voci B e C provenienti dal canale destro. Se volete evitare che i suoni risultino nettamente distinti e desiderate poter registrare più di quattro suoni contemporaneamente, potete usare un registratore come il Tascam Ministudio Porta One.

Il Porta One è un registratore portatile in grado di registrare usando quattro canali invece dei soliti due disponibili su un normale registratore stereo. Il Porta One dispone di un pannello di mixaggio incorporato che consente di registrare i segnali di destra e di sinistra dell'Amiga in uno o più dei quattro canali. Il pannello di mixaggio permette di regolare il volume di ogni singolo canale di registrazione e di fondere i canali destro e sinistro che provengono, ben distinti l'uno dall'altro, dall'Amiga. Inoltre, mentre si registra, è possibile dare l'impressione che uno dei canali provenienti dall'Amiga si muova da sinistra verso destra.

Con quattro canali a disposizione, il Porta One consente di registrare due partiture Amiga sullo stesso nastro e di risentirle poi in stereofonia. Ad esempio, se si vuole realizzare un componimento a otto voci servendosi solo dell'Amiga (magari l'arrangiamento di una sinfonia di Beethoven), si possono inserire quattro voci in una partitura chiamata Beethoven A e quattro voci in una seconda partitura denominata Beethoven B. A quel punto si può registrare Beethoven A in stereo sui canali 1 e 2; a operazione terminata è possibile riavvolgere il nastro e iniziare la registrazione di Beethoven B accanto a Beethoven A sui canali 3 e 4. Riascoltando il tutto, si sentiranno contemporaneamente entrambe le partiture e quindi tutte le otto voci.

Il registratore Porta One offre inoltre la possibilità di sovraincidere due nastri. Ciò significa che è possibile miscelare il contenuto di un nastro qualsiasi con il contenuto di un'altra cassetta, registrando le due fonti su una terza pista; oppure si può aggiungere un nuovo segnale a un canale già registrato, incidendo canale e segnale su un'altra pista. In teoria, attraverso tali operazioni, è possibile aggiungere un numero infinito di

voci alla registrazione. Ad esempio, la sovraincisione può essere utilizzata dall'Amiga per registrare tutte le circa 50 voci necessarie a una riproduzione precisa del "Rito della Primavera" di Strawinsky. Nella pratica, però, ogni sovraincisione deteriora leggermente la qualità della registrazione, e pertanto troppe operazioni di questo tipo potrebbero rendere il suono sibilante e indistinto. Il registratore Porta One dispone di un eccellente dispositivo limitatore di disturbi che consente di eseguire diverse serie di sovraincisioni, per registrare più di 40 voci Amiga senza una distorsione eccessiva.

LA SINCRONIZZAZIONE DELLE PARTITURE

Se si utilizza un mixer o una sovraincisione per combinare le partiture di Deluxe Music, o se si desidera far suonare due Amiga in contemporanea, è importante assicurarsi che le partiture inizino a suonare esattamente nello stesso istante. Il sistema migliore perché ciò avvenga consiste nella creazione di una "introduzione" (lead-in) per uno degli spartiti: un insieme di battute introdotte prima che la partitura venga eseguita, in modo da fornire l'esatto tempo al musicista e permettergli così di avviare il secondo spartito con estrema precisione.

Per creare tale introduzione, inserite all'inizio della partitura una o due misure di uno strumento che suoni a ogni battuta. Per esempio, in una partitura in 4/4 si potrebbero aggiungere come introduzione due misure con sole note semiminime (1/4) e uno strumento a percussione che batte ogni nota. Quando in seguito si riascolterà la partitura, si sentiranno otto battute introduttive di tamburo. La seconda partitura può essere sincronizzata con la prima facendola partire nel punto in cui cadrebbe un'eventuale nona battuta del tamburo di introduzione: in questo modo i due spartiti saranno avviati simultaneamente. Per ottenere buoni risultati sarà necessaria un po' di pratica e di attenzione; in ogni caso, è possibile fermare il registratore e ricominciare qualora si sia persa la battuta.

ALTRI SOFTWARE MUSICALI

Oltre a Deluxe Music, esiste una varietà di programmi musicali per l'Amiga che possiedono caratteristiche e qualità completamente differenti. La possibilità di scelta è vasta, e si può tranquillamente affermare che può essere soddisfatta quasi ogni esigenza specifica, anche la più particolare.

INSTANT MUSIC

Instant Music è un programma della Electronic Arts, semplice ma divertente, perfetto per i musicisti principianti. Possono essere inserite partiture lunghe fino a 64 misure, servendosi di strumenti a suono campionato inclusi nel dischetto e riproducibili a diverse velocità. Le partiture, invece di usare la notazione tradizionale, si presentano come grafici dei toni e del tempo. Viene quindi facilitato il lavoro di coloro che, non sapendo leggere la musica sul pentagramma, possono semplicemente disegnare le voci su un grafico, utilizzando il mouse e il puntatore.

Instant Music consente inoltre all'operatore di eseguire pezzi musicali utilizzando una speciale caratteristica definita "mousejam". Grazie al mousejam Instant Music alza e abbassa i toni a seconda che il mouse venga mosso verso l'alto o verso il basso. È possibile suonare una partitura e utilizzare il mousejam come accompagnamento. Infine si possono ridurre le note e i ritmi ottenibili con il mousejam in modo che, indipendentemente dal movimento del puntatore, i suoni che ne derivano non stonino mai con la partitura di accompagnamento.

SOUNDSCAPE

SoundScape è un programma musicale della Mimetics Corporation costituito in realtà da un insieme di programmi. È in grado di eseguire una varietà di funzioni che lo rendono utile a un numero di utenti abbastanza vasto, tra i quali i musicisti neofiti intenzionati a usare l'Amiga come strumento musicale, i musicisti esperti desiderosi di collegare più sintetizzatori per comporre una musica ricercata, e infine i programmatori esperti alla ricerca di tecniche semplici per aggiungere un sottofondo musicale ai loro programmi.

Il fulcro di SoundScape è un sistema operativo musicale. Quando si carica il programma da disco, questo sistema operativo va a integrarsi al sistema operativo dell'Amiga; i due sistemi funzionano pertanto simultaneamente. Il sistema operativo musicale contiene parecchie routine per comporre ed eseguire partiture, gestire tabelle di forme d'onda e creare musica; queste routine si occupano dell'aspetto musicale nello stesso modo in cui le libraries e devices del sistema operativo dell'Amiga si curano della grafica, dei testi, delle operazioni matematiche e di altre funzioni. Le routine di SoundScape sfruttano al meglio il software dedicato alla gestione del suono presente nel sistema operativo dell'Amiga.

I programmatori esperti possono utilizzare il sistema operativo musicale di SoundScape come una semplice versione ampliata del sistema operativo dell'Amiga, che permette di fare musica senza sobbarcarsi tutto il peso della programmazione.

Dal momento che le routine musicali di SoundScape operano simultaneamente ad altri programmi dell'Amiga, un programmatore che sta realizzando un programma educativo può fornire a SoundScape alcune semplici informazioni sulla musica che desidera per la sua opera, e SoundScape suonerà la musica in sottofondo mentre il programma culturale svolge tranquillamente la sua funzione educativa.

SoundScape mette a disposizione dei non programmatori i suoi programmi che si servono del sistema operativo musicale. Caricando il programma in memoria, si aprono sullo schermo numerose finestre, contenenti ciascuna il proprio programma. Tutti questi programmi operano contemporaneamente, consentendo di produrre musica molto particolare.

Uno dei programmi più semplici offre la possibilità di suonare sulla tastiera dell'Amiga come se fosse una vera e propria tastiera musicale. Altra applicazione piuttosto semplice è l'orologio MIDI, una sorta di metronomo sofisticato che batte il tempo con ritmi e suddivisioni diverse, in modo che i programmi e gli strumenti MIDI a esso collegati vengano sincronizzati dai suoi segnali.

Un altro utilissimo e divertentissimo programma di SoundScape è il produttore di suoni campionati, che consente di modificare gli strumenti a suono campionato creati col campionatore di suoni SoundScape della Mimetics (descritto più avanti in questo capitolo) o con altri programmi come Instant Music o Deluxe Music. Il produttore di suoni campionati può essere usato per selezionare la parte di un suono che andrà a costituire il riverbero, per definire l'inviluppo dell'ampiezza di uno strumento, per trasporre il suono in ottave diverse... A lavoro ultimato, il produttore salva gli strumenti a suono campionato, nuovi o modificati, per consentire all'utente di utilizzarli con altri programmi di SoundScape o con altro software musicale.

Il più ampio di tutti i programmi SoundScape è il registratore MIDI. Tale programma registra note e altri eventi dell'esecuzione musicale realizzata da sintetizzatori collegati all'Amiga attraverso un adattatore MIDI o direttamente alla tastiera dell'Amiga. Il registratore MIDI dispone di un numero illimitato di piste. È possibile registrare musica su ogni pista e poi combinare il suono di singole piste in una pista unica, oppure risentire le registrazioni combinando le varie piste a piacimento attraverso il convertitore MIDI o i canali audio dell'Amiga. Il registratore MIDI può inoltre ricevere e trasmettere su 16 canali differenti; quindi può essere usato per registrare e controllare un massimo di 16 apparecchi MIDI alla volta. Questo programma consente anche di esaminare le informazioni registrate su

una pista qualsiasi, di correggere eventuali errori commessi durante l'esecuzione della musica o di aggiungere nuove note.

Un altro programma SoundScape, il pannello di mixaggio, controlla tutti gli altri programmi che operano simultaneamente. Offre la possibilità di collegare qualsiasi programma SoundScape e canale MIDI a ogni altro programma e canale, fornendo all'utente numerose opportunità. Ad esempio, è possibile usare la tastiera dell'Amiga per registrare la propria esecuzione sul registratore MIDI, oppure si possono collegare i segnali MIDI In all'Amiga stesso per poter suonare gli strumenti a suono campionato immagazzinati nella memoria del computer, usando la tastiera di un sintetizzatore collegato. Le possibilità offerte dal pannello di mixaggio sono sufficienti per tenervi occupati per interi mesi, e quando si considera che la natura modulare di SoundScape semplifica l'aggiunta di altri programmi, queste possibilità diventano addirittura illimitate.

LO STANDARD MUSICALE IFF

Vi ricorderete che nel terzo capitolo abbiamo parlato dello standard IFF, creato dalla Electronic Arts e dalla Commodore per facilitare il trasferimento di dati da un programma a un altro. Lo standard IFF per la grafica consente di scambiare immagini fra programmi quali Deluxe Paint e Aegis Images. Esistono anche standard IFF applicabili alla musica: uno per le partiture e un altro per gli strumenti a suono campionato.

Se un programma utilizza lo standard IFF per salvare le partiture e gli strumenti, entrambi gli elementi possono poi essere introdotti in altri programmi conformi allo stesso standard. Per esempio, sia SoundScape sia il campionatore di suoni salvano gli strumenti servendosi dello standard IFF; pertanto, qualsiasi strumento venga creato con questi programmi può venire utilizzato in Deluxe Music e Instant Music.

HARDWARE MUSICALE PER L'AMIGA

Così come si può aggiungere hardware all'Amiga per potenziarne le capacità grafiche, è anche possibile introdurre nuovi elementi per migliorarne le capacità sonore e di composizione musicale. I paragrafi seguenti descrivono parte di questo hardware, che copre un'intera gamma di strumenti: da semplici altoparlanti esterni a reti di sintetizzatori.

ALTOPARLANTI

Gli altoparlanti dei monitor per computer, in massima parte, non sono molto migliori dei piccoli altoparlanti dal suono metallico di cui è dotato un televisore di media qualità. Se vi è mai capitato di sentire il suono distorto della musica proveniente da un televisore medio, capirete certamente che conviene dotare il vostro Amiga di altoparlanti ad alta fedeltà, soprattutto se pensate di utilizzarlo per musica di ogni genere sfruttando al meglio le capacità multivocali e sonore di cui è dotata la macchina.

Collegamento a un sistema stereofonico

Se possedete un buon impianto stereo, potete collegare l'Amiga all'amplificatore del sistema in modo da sentire i suoni del computer riprodotti attraverso le casse stereo. L'operazione è semplicissima; basta collegare l'Amiga come una qualunque piastra di registrazione, dal momento che esso produce un segnale audio lineare di potenza pari alla maggior parte dei registratori.

Per collegare l'Amiga a un apparecchio stereo, assicuratevi prima che entrambi siano spenti, dopodiché collegate le porte audio del computer a una delle coppie di entrate stereo sul retro dell'amplificatore utilizzando un cavo a due vie, nello stesso modo in cui collegheresteste l'Amiga a un registratore. Le entrate dell'amplificatore possono essere quelle indicate per il registratore, il lettore di compact disc, il videoregistratore o altre apparecchiature ausiliarie. L'unica entrata che non deve assolutamente essere usata è quella per il piatto dello stereo; infatti questa entrata è predisposta per ricevere segnali molto più deboli di quelli emessi dall'Amiga e, pertanto, il suo utilizzo provocherebbe una distorsione del suono o addirittura la rottura degli altoparlanti nel caso in cui si alzasse troppo il volume dell'amplificatore.

Dopo che il collegamento è stato effettuato, accendete entrambi i sistemi, caricate il programma musicale ed eseguite una partitura. Aumentate gradatamente il volume dell'amplificatore; se gli altoparlanti emettono un suono assordante, risulta evidente che il segnale dell'Amiga è troppo forte. Ripetete quindi il collegamento provando un'altra coppia di entrate.

Altoparlanti preamplificati

Se non si intende collegare l'Amiga a un apparecchio stereo, è possibile ricorrere all'acquisto di un paio di altoparlanti autonomi muniti di amplificatore incorporato. Gli altoparlanti

preamplificati non devono essere inseriti in un impianto sonoro: è sufficiente connetterli alle porte audio dell'Amiga e accenderli. Questi altoparlanti sono generalmente piccoli e possono essere piazzati sulla propria scrivania, uno a destra e l'altro a sinistra dell'Amiga per ottenere un suono stereo.

Il modello più comune è quello per registratori portatili poco potenti. Sfortunatamente, la maggior parte di questi altoparlanti non si adattano all'Amiga perché sono predisposti per ricevere il segnale proveniente dal jack per la cuffia del registratore, segnale già amplificato e quindi più potente di un segnale lineare. Gli altoparlanti destinati alla ricezione di un segnale inviato da un jack per cuffie non possiedono sufficiente capacità di amplificazione per funzionare anche con l'Amiga.

Alcuni però, progettati per essere utilizzati con un sintetizzatore, hanno un'entrata lineare e quindi dispongono di un alto livello di amplificazione, emettendo solitamente un suono nitido. L'altoparlante Casio AS-20 è un ottimo esempio: possiede un equalizzatore grafico incorporato, manopole che consentono di regolare il suono a piacimento, un jack per cuffie e due entrate separate.

L'acquisto degli altoparlanti

Quando andate alla ricerca di un paio di altoparlanti per l'Amiga, è importante verificarne la compatibilità. Richiedete una dimostrazione pratica collegandoli a una piastra di registrazione o a un lettore di compact disc. Assicuratevi che gli altoparlanti vengano collegati non al jack per la cuffia, bensì ai jack line-out. Poiché registratori e lettori laser emettono un segnale di intensità pari a quello dell'Amiga, gli altoparlanti funzionanti con questi apparecchi dovrebbero essere quindi perfetti anche per l'Amiga.

Registratori portatili

Per ottenere un buon suono dall'Amiga potete anche utilizzare un registratore portatile a batteria dotato di amplificatore e altoparlanti stereo incorporati. Molti di questi apparecchi dispongono di entrate in cui potete inserire il cavo proveniente dall'Amiga. I registratori portatili offrono un ulteriore vantaggio: possono registrare la musica creata con l'Amiga.

I CAMPIONATORI DI SUONI

Alcuni degli strumenti migliori tra i programmi musicali finora menzionati sono gli strumenti a suono campionato. Tali strumenti vengono creati da un campionatore di suoni (detto anche digitalizzatore audio), ovvero un dispositivo che converte segnali analogici provenienti da microfoni, registratori e altre sorgenti sonore in una tavola di forma d'onda digitale.

Per la realizzazione di propri strumenti a suono campionato, è possibile acquistare un campionatore di suoni e iniziare a collezionare le proprie creazioni sonore. Molti campionatori sono progettati appositamente per l'Amiga. Alcuni possiedono microfoni incorporati, altri utilizzano campioni di suoni stereo. La ricerca di software che definisca i campioni di suoni e li salvi su dischetto come strumenti a suono campionato riveste un ruolo di primaria importanza.

Uno dei migliori (e più economici) campionatori presenti sul mercato è il campionatore di suoni SoundScape della Mimetics Corporation. Si tratta di un piccolo campionatore stereo che deve essere inserito nella porta del joystick; questa caratteristica costituisce già un vantaggio rispetto ai campionatori che si inseriscono nel connettore di espansione, impedendo l'aggiunta di altre periferiche. Sebbene non sia dotato di microfono incorporato, il campionatore SoundScape possiede un jack per microfono e una serie di jack stereo lineari che consentono di collegare il proprio microfono o un'altra sorgente sonora.

Il campionatore di suoni SoundScape viene venduto unitamente a programmi che offrono la possibilità di creare la tavola di una forma d'onda, perfezionarla e salvarla su disco. Il sistema utilizzato per produrre i suoni è identico a quello adottato dal programma SoundScape descritto nella precedente sezione dedicata al software. Il software permette anche di controllare l'Amiga tramite un adattatore MIDI; è pertanto possibile "suonare" l'Amiga e i suoi suoni campionati con un sintetizzatore esterno collegato al computer.

L'INTERFACCIA MIDI

Come ricorderete dallo scorso capitolo, per MIDI si intende uno standard utilizzato per la fabbricazione di sintetizzatori che consente lo scambio di informazioni fra due o più apparecchi. Per trasformare i segnali provenienti dall'Amiga in segnali MIDI, è necessario collegare un'interfaccia MIDI alla porta seriale dell'unità centrale. Le aziende che vendono

convertitori MIDI per l'Amiga sono piuttosto numerose.

Qualsiasi adattatore dovrebbe essere adatto alla porta seriale dell'Amiga e munito di tutte e tre le connessioni: MIDI In, MIDI Out e MIDI Thru. Ricordatevi che alcune case produttrici vendono l'adattatore MIDI unitamente ai loro programmi software, mentre altre vendono il solo adattatore a prezzi piuttosto elevati.

SINTETIZZATORI MIDI

Una volta in possesso di un adattatore MIDI, è possibile espandere l'Amiga con ben 16 sintetizzatori e altre apparecchiature musicali munite di interfacce MIDI. Se progettate di collegare all'Amiga un sintetizzatore da usare con il software di registrazione MIDI, il sintetizzatore dovrebbe essere dotato perlomeno di jack MIDI In e MIDI Out per inviare e ricevere informazioni dal software. Intendendo collegare più di un sintetizzatore, è importante assicurarsi che vi sia anche una porta MIDI Thru su ognuno di essi.

Esiste una vasta gamma di sintetizzatori MIDI, da quelli professionali del costo di decine di milioni, a quelli molto simili a giocattoli e acquistabili per poche centinaia di migliaia di lire. Uno dei migliori sintetizzatori disponibili a prezzo ragionevole è il Casio CZ-101.

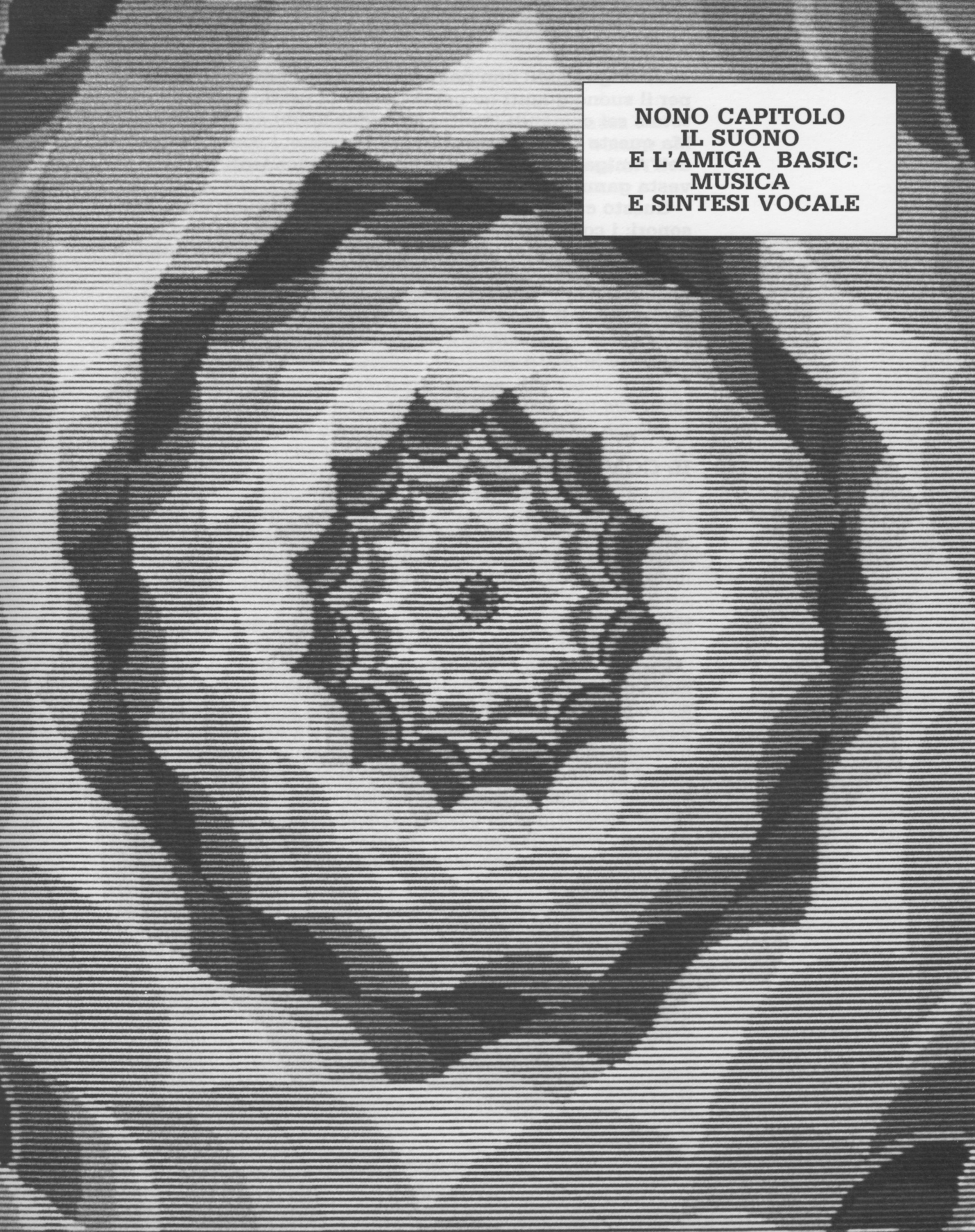
Si tratta di un sintetizzatore digitale completo di un set di 32 strumenti che mette a disposizione dell'utente anche comandi per creare strumenti nuovi e memorizzarli su piccole cartucce RAM inseribili nella parte posteriore del sintetizzatore stesso.

Il CZ-101 dispone di una tastiera a quattro ottave in grado di suonare fino a otto note contemporaneamente, e di un regolatore che consente di alzare o abbassare la tonalità mentre si suona. Sfortunatamente la tastiera è più piccola di quelle normali e può quindi causare problemi di adattamento. Un altro sintetizzatore da prendere in considerazione è il Casio CZ-1000, identico al CZ-101 fatta eccezione per la tastiera di dimensioni normali e il prezzo decisamente superiore.

Il CZ-101 è munito di porte MIDI In e MIDI Out che consentono di utilizzarlo con il software di registrazione MIDI. L'abbinamento con quest'ultimo è ulteriormente completato da due particolari modi MIDI dell'apparecchio: il CZ-101 può essere usato come sintetizzatore singolo in grado di suonare fino a otto note simultaneamente su un unico canale, oppure, in un secondo modo, come quattro sintetizzatori separati che suonano su quattro canali MIDI distinti.

Il secondo modo consente di selezionare dalla memoria quattro diversi strumenti, uno per canale. Ogni strumento è monofonico, può cioè far sentire solo una nota alla volta e non interi accordi. Per adoperare questo modo speciale, il registratore MIDI può inviare segnali attraverso i quattro canali separati allo scopo di suonare una partitura a quattro voci utilizzando strumenti simili. Poiché tutti i 16 canali MIDI sono contenuti in un unico cavo, è sufficiente un'unica connessione con la porta MIDI In. La maggior parte dei sintetizzatori non consente l'uso di più strumenti contemporaneamente; il CZ-101 offre quindi una versatilità abbastanza inconsueta e decisamente interessante.

In questo capitolo vi sono stati forniti suggerimenti utili per immettere le vostre partiture in Deluxe Music in modo facile e veloce. La seconda parte è stata invece dedicata alla presentazione del software e dell'hardware che può aprire nuovi orizzonti al musicista che lavora con l'Amiga. Il prossimo capitolo spiegherà il funzionamento dei comandi dell'Amiga BASIC riservati alla creazione di musica e di sintesi vocale.



NONO CAPITOLO
IL SUONO
E L'AMIGA BASIC:
MUSICA
E SINTESI VOCALE

L'Amiga BASIC non possiede tanti comandi e funzioni per il suono quanti ne possiede per la grafica: esistono in totale sei comandi (contando anche BEEP) e una funzione. Ma questo non vi deve trarre in inganno: i comandi sonori dell'Amiga BASIC offrono prestazioni eccellenti e una vasta gamma di effetti speciali.

Questo capitolo prenderà in esame due tipi di comandi sonori: i comandi che creano musica ed effetti sonori e i comandi, o meglio il comando e la funzione, che simulano la voce umana. Oltre a spiegare il funzionamento dei comandi, sarà illustrato il modo in cui utilizzarli in programmi di una certa estensione. Infine verranno fornite le nozioni necessarie a creare le scale musicali e a eseguire le partiture.

IL COMANDO BEEP

Il comando BEEP rappresenta il modo più facile per avvicinarsi alle capacità sonore dell'Amiga. È un comando estremamente semplice, e utilizza la seguente sintassi:

```
BEEP
```

Quando un programma esegue il comando BEEP, il monitor lampeggia un istante e l'altoparlante emette un breve suono. Il suono (che per comodità definiremo beep) proviene dalla porta audio sinistra posta sul retro dell'unità centrale dell'Amiga. La frequenza del beep è di 880 hertz, corrisponde cioè al LA del pianoforte situato nella seconda ottava al di sopra del DO centrale. Il beep utilizza una qualsiasi delle forme d'onda del canale audio 0, quindi è possibile modificarne il timbro con il comando WAVE. Per saperne di più sui canali audio e sui cambiamenti di timbro, leggete i paragrafi che seguono, dedicati ai comandi SOUND e WAVE.

Non essendo possibile modificare la lunghezza e il tono del suono emesso dal comando BEEP, quest'ultimo viene usato raramente per la musica e gli effetti sonori. Viene invece più spesso utilizzato per segnalare all'utente la fine di una lunga operazione BASIC. BEEP richiama l'attenzione dell'operatore evitandogli la noia di tenere sotto continuo controllo l'Amiga.

IL COMANDO SOUND

Il comando SOUND può suonare una nota o una pausa e regolare tre proprietà del suono: frequenza, durata e volume. Inoltre controlla il canale audio di provenienza delle note. La sua sintassi è la seguente:

SOUND *frequenza, durata, volume, canale audio*

La frequenza, che può essere indicata da qualsiasi valore compreso fra 20 e 15000, specifica il numero di cicli al secondo (cps) che definisce il tono della nota. La durata, rappresentata da valori compresi fra 0 e 77, regola la lunghezza della nota su valori variabili da 0 a 4,23 secondi. Il volume è rappresentato da un numero compreso fra 0 e 255. Inserendo il valore 0, il volume è nullo (silenzio), mentre a 255 il volume è al massimo. Infine, il canale audio deve essere specificato con un numero compreso fra 0 e 3 che indichi il canale a cui si vuole assegnare la nota.

I valori del volume e del canale audio sono opzionali. Se non vengono inclusi, il valore del volume sarà automaticamente di 127 (una via di mezzo fra il silenzio e il volume massimo), mentre il canale utilizzato attraverso la porta audio di sinistra sarà il canale numero 0.

REGOLAZIONE DELLA FREQUENZA

Se si ha una certa dimestichezza con i toni misurati in cicli al secondo, la regolazione della tonalità di una nota tramite il comando SOUND risulterà molto semplice. È sufficiente inserire la frequenza del tono desiderato. Se si trova invece meno complicato utilizzare le note indicandole con i loro nomi (DO, RE, MI, FA, SOL, LA, SI), un breve studio della relazione fra frequenza e tono faciliterà la comprensione della tecnica necessaria per regolare una nota sulla tonalità desiderata.

Allo scopo di differenziare i toni delle varie ottave esistenti, quest'ultime vengono solitamente numerate da 0 a 10, in modo da poter indicare l'ottava in cui cade un determinato tono. Pertanto MI_0 è il MI dell'ottava 0, la più bassa, e il $SI\flat_{10}$ è il SI♭ della decima ottava, ovvero l'ottava più alta. Questo sistema dell'Amiga BASIC non deve essere però confuso con i sistemi usati per definire i nomi dei toni nei programmi di applicazione musicale, in cui la numerazione delle ottave talvolta incomincia da un'ottava più alta e non corrisponde quindi alla descrizione presentata in questo capitolo.

La gamma delle frequenze utilizzabili col comando SOUND varia da 20 a 15000 cps, ovvero da MI_0 (il terzo MI al di sotto dell'ultima nota della chiave chiave di basso) a $SI\flat_9$ (il quinto SI♭ al di sopra della chiave di violino). Si tratta di una scala di

otto ottave e mezzo, paragonata a quella di sei ottave e mezzo di un pianoforte.

Per avere un'idea della relazione fra frequenze e toni specifici, prendete in considerazione alcuni esempi piuttosto noti. Il DO centrale del pianoforte è un DO_4 e ha una frequenza di 261,63 cps. Il LA immediatamente inferiore a questo DO è un LA_4, la nota che le orchestre utilizzano per accordare gli strumenti. I musicisti si riferiscono a essa come al LA440, e quindi non stupisce il fatto che la sua frequenza sia di 440 cps.

Figura 9-1.

La lista dei toni utilizzabili dal comando SOUND con le rispettive frequenze.

Tono	Frequenza	Tono	Frequenza	Tono	Frequenza
MI_0	20.602	SOL_3	196.00	$LA\#_6$	1864.7
FA_0	21.827	$SOL\#_3$	207.65	SI_6	1975.5
$FA\#_0$	23.125	LA_3	220.00		
SOL_0	24.500	$LA\#_3$	233.08	DO_7	2093.0
$SOL\#_0$	25.957	SI_3	246.94	$DO\#_7$	2217.5
LA_0	27.500			RE_7	2349. 3
$LA\#_0$	29.13	DO_4	261.63	$RE\#_7$	2489.0
SI_0	30.868	$DO\#_4$	277.185	MI_7	2637.0
		RE_4	293.66	FA_7	2793.8
DO_1	32.703	$RE\#_4$	311.13	$FA\#_7$	2960.0
$DO\#_1$	34.648	MI_4	329.63	SOL_7	3136.0
RE_1	36.708	FA_4	349.23	$SOL\#_7$	3322.4
$RE\#_1$	38.891	$FA\#_4$	369.99	LA_7	3520.0
MI_1	41.203	SOL_4	392.00	$LA\#_7$	3729.3
FA_1	43.654	$SOL\#_4$	415.30	SI_7	3951.1
$FA\#_1$	46.249	LA_4	440.00		
SOL_1	48.999	$LA\#_4$	466.16	DO_8	4186.0
$SOL\#_1$	51.913	SI_4	493.88	$DO\#_8$	4434.9
LA_1	55.000			RE_8	4698.6
$LA\#_1$	58.270	DO_5	523.25	$RE\#_8$	4978.0
SI_1	61.735	$DO\#_5$	554.37	MI_8	5274.0
		RE_5	587.33	FA_8	5587.7
DO_2	65.406	$RE\#_5$	622.25	$FA\#_8$	5919.9
$DO\#_2$	69.296	MI_5	659.26	SOL_8	6271.9
RE_2	73.416	FA_5	698.46	$SOL\#_8$	6644.9
$RE\#_2$	77.782	$FA\#_5$	739.99	LA_8	7040.0
MI_2	82.407	SOL_5	783.99	$LA\#_8$	7458.6
FA_2	87.307	$SOL\#_5$	830.61	SI_8	7902.1
$FA\#_2$	92.499	LA_5	880.00		
SOL_2	97.999	$LA\#_5$	932.33	DO_9	8372.0
$SOL\#_2$	103.83	SI_5	987.77	$DO\#_9$	8869.8
LA_2	110.00			RE_9	9397.3
$LA\#_2$	116.54	DO_6	1046.5	$RE\#_9$	9956.1
SI_2	123.47	$DO\#_6$	1108.7	MI_9	10548.1
		RE_6	1174.7	FA_9	11175.3
DO_3	130.81	$RE\#_6$	1244.5	$FA\#_9$	11839.8
$DO\#_3$	138.59	MI_6	1318.5	SOL_9	12543.9
RE_3	146.83	FA_6	1396.9	$SOL\#_9$	13289.8
$RE\#_3$	155.56	$FA\#_6$	1480.0	LA_9	14080.0
MI_3	164.81	SOL_6	1568.0	$LA\#_9$	14917.2
FA_3	174.61	$SOL\#_6$	1661.2		
$FA\#_3$	185.00	LA_6	1760.0		

Raddoppiando la frequenza di un tono, si ottiene lo stesso tono un'ottava più alto, dimezzando la frequenza si ottiene viceversa lo stesso tono un'ottava più sotto. Ad esempio, se si raddoppia la frequenza del LA_4 di 440 cps, ne derivano 880 cps, corrispondenti al LA_5, il LA dell'ottava superiore. Viceversa, 220 cps equivalgono alla metà di 440, quindi al LA_3, un'ottava più sotto rispetto al LA_4.

La Figura 9-1 mostra la lista dei toni ottenibili con SOUND affiancati dalle rispettive frequenze. Questo elenco dei toni costituisce una scala temperata, ovvero un modello utilizzato dalla maggior parte dei musicisti, accordato in modo tale che il LA_4 equivalga a 440 cps.

Per poter sentire tutti questi toni è necessario collegare l'Amiga a un buon sistema audio. Se l'utente si servirà soltanto del monitor, probabilmente non sentirà alcun tono inferiore ai 100 cps e superiore a 8000 cps, a meno che non cambi la forma d'onda del tono col comando WAVE (discusso più avanti in questo capitolo).

REGOLAZIONE DELLA DURATA

I valori di durata del comando SOUND possono variare da 0 a 77. Quanto più il numero è elevato, tanto maggiore è la durata della nota. Se il valore è 0, la nota è priva di lunghezza; con un valore pari a 1, la nota dura poco più di 1/20 di secondo. La nota dura esattamente un secondo se il valore introdotto è 18.2, mentre il valore massimo di 77 fa durare un suono circa 4,23 secondi. Ad esempio, questo comando suona un LA440 per un secondo:

```
SOUND 440, 18.2
```

Se si calcola la lunghezza in secondi della nota che si vuole suonare e la si moltiplica per 18,2, si ottiene immediatamente il valore che dovete introdurre nel comando SOUND per avere l'effetto desiderato. Ad esempio, un suono di mezzo secondo ha come valore di durata 9.1. Per prolungare la durata di un suono oltre i 4,23 secondi, è possibile servirsi di due o più comandi SOUND che utilizzino gli stessi valori di tono, volume e canale audio. I comandi opereranno congiuntamente per creare una nota lunga. I due comandi seguenti suonano un FA_4 della durata di sette secondi:

```
SOUND 349.23, 72.8
SOUND 349.23, 54.6
```

Il primo comando suona una nota di 4 secondi, il secondo completa gli altri 3. Poiché nei comandi non sono specificati né il volume, né il canale audio, entrambe le note hanno un volume pari a 127 e si avvalgono del canale 0.

REGOLAZIONE DEL VOLUME

La scelta del volume per il comando SOUND consiste semplicemente nel selezionare un numero compreso fra 0 e 255. Se si seleziona 0, non si potrà sentire la nota desiderata e, pertanto, la frequenza prefissata non avrà alcuna importanza. Il silenzio durerà per tutto il tempo stabilito e quindi sarà possibile utilizzare il volume 0 per creare una pausa. Ad esempio, il secondo dei tre comandi SOUND riportati qui di seguito crea una pausa di silenzio di mezzo secondo fra due note suonate al massimo volume:

```
SOUND 440, 18.2, 255
SOUND 440, 9.1, 0
SOUND 392, 18.2, 255
```

I valori di volume compresi fra 0 e 255 possono essere utilizzati per suonare la stessa nota a volumi diversi. È sufficiente ricordare che il volume è determinato anche dalla regolazione del monitor o del sistema stereo collegato all'Amiga. Se abbassate il volume del sistema della metà, gli stessi comandi usati in precedenza produrranno questa volta suoni a volume dimezzato.

SCELTA DEL CANALE AUDIO

Per suonare una nota creata col comando SOUND è possibile scegliere uno qualsiasi dei quattro canali audio interni dell'Amiga. Utilizzando quattro diversi comandi SOUND, uno per canale, si possono suonare contemporaneamente quattro note. I quattro comandi SOUND che seguono riproducono un accordo di DO maggiore:

```
SOUND 261.63, 20, 255, 0
SOUND 329.63, 20, 255, 1
SOUND 392, 20, 255, 2
SOUND 523.25, 20, 255, 3
```

I suoni emessi dai quattro canali vengono mixati prima di giungere all'altoparlante esterno passando attraverso le due porte audio sul retro dell'unità centrale. I canali 0 e 3 fuoriescono dalla porta sinistra, i canali 1 e 2 da quella destra. Se l'Amiga è collegato a due altoparlanti stereo, si può fare in modo che il suono di una nota venga emesso solo da uno dei due altoparlanti, semplicemente indicando nel comando SOUND il canale appropriato.

Ogni canale interno ha una propria forma d'onda che determina il timbro delle note create col comando SOUND. La forma d'onda può essere cambiata utilizzando il comando WAVE. La selezione di forme d'onda diverse per ciascuno dei canali audio

consente di scegliere il timbro della nota suonata dal comando SOUND; per farlo, è sufficiente specificare il canale contenente la forma d'onda desiderata.

LA SINCRONIZZAZIONE DEI SUONI

Come risulta chiaro dall'ultimo esempio, i comandi SOUND consentono all'utente di creare note contemporaneamente su tutti e quattro i canali audio interni dell'Amiga. Tuttavia, ottenere che tutte e quattro le note inizino simultaneamente può comportare delle difficoltà, soprattutto se fra i comandi SOUND sono stati inseriti altri comandi BASIC. Dal momento che l'Amiga BASIC esegue un comando alla volta, è possibile che si verifichino dei ritardi nel passaggio da un comando all'altro e, di conseguenza, venga a mancare la sincronizzazione fra le note. L'Amiga BASIC dispone però di due comandi che facilitano la sincronizzazione: SOUND WAIT e SOUND RESUME.

I COMANDI SOUND WAIT E SOUND RESUME

SOUND WAIT viene utilizzato insieme a SOUND RESUME per sincronizzare i canali audio dell'Amiga. È possibile quindi assicurarsi che le note di una partitura a più voci vengano eseguite contemporaneamente ed evitare il rischio di un'eventuale stonatura. Con SOUND WAIT, i suoni prodotti utilizzando il comando SOUND non riescono a raggiungere l'altoparlante e rimangono "in pausa" nell'Amiga, come se fossero conservati in un contenitore. Ogni canale audio possiede il proprio contenitore per immagazzinare un massimo di dodici elementi, note o pause. Per agire su tutti e quattro i contenitori (code) è sufficiente un solo comando SOUND WAIT. Se si cerca di immagazzinare più delle dodici note e pause consentite (dodici comandi SOUND inviati sullo stesso canale) si commette un errore segnalato dal computer col messaggio "out of memory". Il comando SOUND WAIT ha una sintassi semplicissima:

```
SOUND WAIT
```

Quando l'Amiga BASIC incontra questo comando, tutti i suoni creati dai comandi SOUND successivi vengono trattenuti nei contenitori, fino a quando non giungerà un comando SOUND RESUME a liberarli. SOUND RESUME invia i suoni dai contenitori agli altoparlanti e permette l'esecuzione simultanea di tutte le note. I contenitori si liberano automaticamente

quando le note vengono suonate. La sintassi di SOUND RESUME è semplicemente:

```
SOUND RESUME
```

Per capire il funzionamento di questi due comandi, provate a far girare il seguente programma, che crea cinque note per ciascuno dei tre canali audio 0, 1 e 2. Subito dopo i comandi SOUND per il primo canale è stato introdotto un ciclo FOR...NEXT che aumenta la durata dell'intervallo prima di passare ai comandi del secondo canale. L'utente distinguerà chiaramente una pausa fra le voci. Alla fine del programma si ha inoltre un ciclo infinito che impedisce la visualizzazione della finestra List; la sua apparizione potrebbe far ritardare uno dei canali audio e quindi ostacolare la sincronizzazione delle quattro voci.

Inserite ora il programma e fatelo girare, senza però includervi i comandi SOUND WAIT e SOUND RESUME:

```
SOUND WAIT
SOUND 261.63, 12, 255, 0
SOUND 196, 12, 255, 0
SOUND 261.63, 12, 255, 0
SOUND 196, 12, 255, 0
SOUND 261.63, 48, 255, 0
SOUND 261.63, 12, 255, 1
SOUND 293.66, 12, 255, 1
SOUND 329.63, 12, 255, 1
SOUND 293.66, 12, 255, 1
SOUND 261.63, 48, 255, 1
FOR t = 0 TO 1000: NEXT t
SOUND 329.63, 12, 255, 2
SOUND 349.23, 12, 255, 2
SOUND 392, 12, 255, 2
SOUND 349.23, 12, 255, 2
SOUND 329.63, 48, 255, 2
SOUND RESUME
Loop: GOTO Loop
```

A questo punto sentirete una voce (il canale 2) iniziare e finire molto più tardi rispetto alle altre due. Ora inserite i comandi SOUND WAIT e SOUND RESUME e fate girare nuovamente il programma. Tutte le voci inizieranno e finiranno contemporaneamente, poiché SOUND WAIT blocca tutte le note e le mette una in fila all'altra nei contenitori fino a quando SOUND RESUME permette di suonarle contemporaneamente.

IL COMANDO WAVE

Il comando WAVE può essere utilizzato per creare una nuova tavola di forma d'onda per ciascuno dei quattro canali audio. Cambiando la tavola di un canale, viene a modificarsi il timbro di qualsiasi suono emesso attraverso quel canale. WAVE usa la seguente sintassi:

```
WAVE canale audio, nome dell'array
```

Il canale audio va indicato con un numero intero compreso fra 0 e 3 che specifichi appunto il canale a cui si intende assegnare la nuova forma d'onda. Il secondo elemento deve invece essere il nome dell'array che contiene i dati della nuova tavola. L'array deve essere creato prima dell'utilizzo del comando WAVE.

È possibile sostituire il nome dell'array con l'indicazione *SIN*, per fare in modo che WAVE crei automaticamente una forma d'onda sinusoidale nel canale audio specificato. Tutti i canali usano una forma d'onda sinusoidale come forma prestabilita, sempre che l'utente non la modifichi con un comando WAVE. Inserendo SIN in un comando WAVE, si restituisce al canale audio la sua forma d'onda di default.

CREAZIONE DI UNA TAVOLA DI FORMA D'ONDA

Prima di creare la tavola vera e propria, la prima cosa da fare è dimensionare un array. Il comando WAVE richiede l'impiego di un array intero; array di altro tipo genereranno un errore di *Type Mismatch*. È necessario che l'array contenga almeno 256 elementi, e può essere più lungo se si desidera definire una forma d'onda maggiormente dettagliata. Ad esempio, il seguente comando dimensiona un array di numeri interi chiamato *form%*, che include 256 elementi (da 0 a 255). Il % sta a indicare che si tratta di un array intero di tipo breve (2 byte).

```
DIM form%(255)
```

La seconda operazione consiste nell'assegnare i dati della forma d'onda agli elementi dell'array. Ciascuno dei 256 elementi, necessariamente numeri interi compresi fra -128 e 127, è disposto in ordine crescente e descrive, da sinistra a destra, la sagoma dell'onda posta in un grafico. Per capire meglio, osservate la Figura 9-2, che mostra una forma d'onda inserita in un grafico e l'ingrandimento della sua sezione iniziale, in cui sono riportati i primi 6 valori della tavola della forma d'onda. Questi ultimi vengono assegnati ai primi 6 elementi dell'array, seguiti dagli altri 250 elementi necessari a definire l'intera forma d'onda.

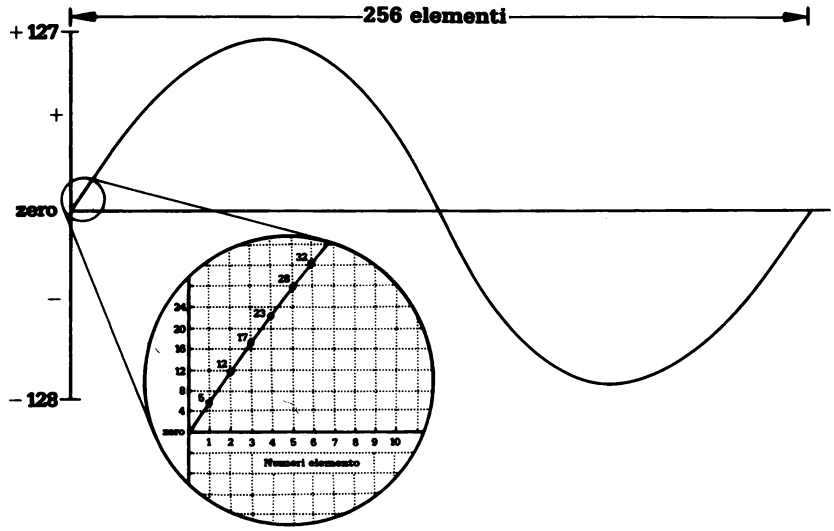


Figura 9-2.

Il grafico di un'onda sonora completa con un ingrandimento della sezione iniziale che mostra i primi sei valori della tavola di quest'onda.

Per associare agli elementi dell'array i valori della forma d'onda si può ricorrere a un ciclo FOR...NEXT che legga, per ogni elemento dell'array, un valore ricavato da un comando DATA letto con un comando READ; oppure si può creare un ciclo FOR...NEXT con una formula matematica che ricava i valori della forma d'onda via via che il ciclo progredisce.

Lettura dei dati della forma d'onda contenuti nell'array

Per leggere i dati della forma d'onda contenuti nell'array con i comandi READ e DATA, occorre innanzitutto avere almeno 256 valori relativi alla tavola. Un sistema per ottenerli consiste nel disegnare dapprima la forma d'onda su un foglio di carta millimetrata e, in seguito, misurare l'ampiezza dell'onda in 256 punti differenti posti lungo l'asse delle ascisse (orizzontale), come indicato dalla Figura 9-2 (nella pagina precedente). È importante graduare l'asse delle coordinate (verticale), allo scopo di evitare che la forma d'onda superi il numero 127 o scenda al di sotto di -128. I valori vanno poi inseriti in una serie di comandi DATA secondo l'ordine di lettura, da sinistra a destra.

I seguenti comandi creano un ciclo FOR...NEXT che opera 256 volte e ogni volta legge un valore della forma d'onda nell'array *form%*. I comandi DATA successivi al ciclo FOR...NEXT contengono i dati della forma d'onda (naturalmente non li abbiamo elencati tutti):

```
DIM form%(255)
FOR i = 0 TO 255
    READ form%(i)
NEXT i
DATA -112, -112, -111, -110, -108, -105, -103, -94, -83 ecc.
```

Calcolo di forme d'onda semplici

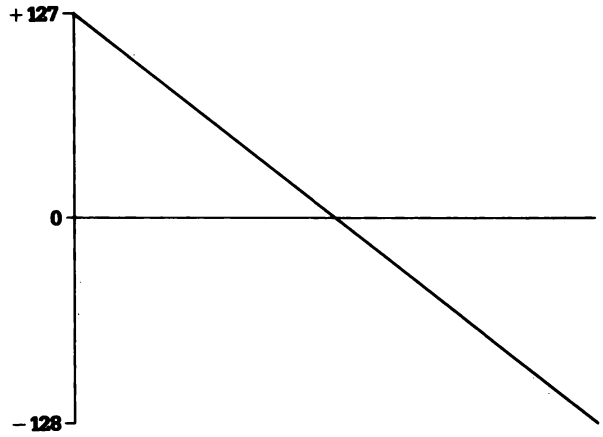
Una tecnica di gran lunga più breve per creare una tavola di forma d'onda consiste nel ricavare la forma d'onda con una semplice formula all'interno di un ciclo FOR...NEXT. Questo sistema si adatta perfettamente alle forme d'onda non complesse, come le onde quadre, le onde a dente di sega e quelle a impulso. Esempi di tutte queste forme d'onda sono riportati nella Figura 9-4 (riportata a pagina 251). I comandi seguenti indicano come creare una forma d'onda a dente di sega:

```
DIM form%(255)
FOR i = 0 TO 255
    form%(i) = 127 - i
NEXT i
```

Questo breve ciclo produrrebbe la forma d'onda della Figura 9-3 se *form%* fosse applicato utilizzando il comando WAVE.

Figura 9-3.

Una forma d'onda a dente di sega può essere creata con una formula semplicissima.



Sono riportate qui di seguito altre formule semplici e passi di programma in grado di creare forme d'onda meno complesse con timbri distinti. È sufficiente inserirle nello stesso ciclo sopra indicato, sostituendo la formula dell'onda a dente di sega con la formula desiderata. La Figura 9-4 riporta le onde ottenute con le formule, fatta eccezione per la forma d'onda casuale, che appare diversa ogni volta che la si utilizza.

Forma d'onda quadra:

```
IF i < 128 THEN form%(i) = 127 ELSE form%(i) = -128
```

Forma d'onda triangolare:

```
IF i < 128 THEN form%(i) = (2*i) - 128 ELSE form%(i) = 383 - (2*i)
```

Forma d'onda a impulso:

```
IF i< 5 THEN form%(i) = 127 ELSE form%(i) = -128
```

Forma d'onda casuale:

```
form%(i) = 127 - INT(256*RND)
```

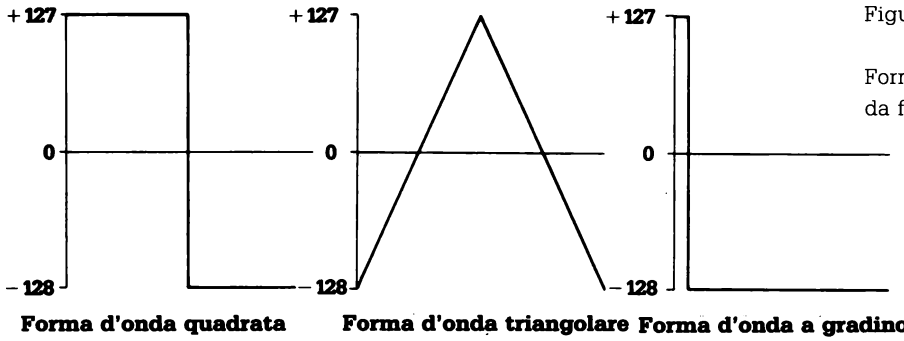


Figura 9-4.

Forme d'onda generate da formule elementari.

È indubbiamente possibile ricavare altre formule che creino forme d'onda con timbri distinti. Perché una formula sia valida è necessario che i suoi risultati siano sempre numeri interi, non inferiori a -128 o superiori a 127. È inoltre importante assicurarsi che almeno alcuni valori si avvicinino ai limiti dell'array di numeri interi, poiché altrimenti il volume del suono non sarebbe molto alto. Se ad esempio si crea una forma d'onda racchiusa entro i limiti 6 e -3, risulterà difficile sentirla indipendentemente dalla regolazione del volume dell'altoparlante del monitor.

ASSEGNAZIONE DI UNA TAVOLA DI FORMA D'ONDA A UN CANALE AUDIO

Dopo aver creato una tavola e averla assegnata a un array, non sarà difficile attribuirla a un canale audio. È sufficiente specificare il canale e il nome dell'array in un comando WAVE. Per ascoltare la forma d'onda, basta utilizzare un comando SOUND, e suonare una nota usando il canale audio in questione. Il breve programma seguente crea una forma d'onda a impulso nel canale 0; in seguito esegue, al massimo volume, un LA_4 della durata di un secondo, cosicché si può sentire la forma d'onda prodotta.

```
DIM form%(255)
FOR i = 0 TO 255
    IF i < 5 THEN form%(i) = 127 ELSE form%(i) = -128
NEXT i
WAVE 0, form%
SOUND 440, 18.2, 255, 0
```

Se l'operatore desidera sentire altre forme d'onda, può provare a sostituire nel programma la formula dell'onda sopracitata con alcune delle altre formule.

LA CREAZIONE DI SCALE MUSICALI

Se si utilizzano comandi SOUND individuali per creare una serie di note, la ricerca di un tono e l'uso della giusta frequenza per ogni nota risulteranno alquanto noiosi. Tale operazione può essere semplificata considerevolmente con la realizzazione di un array che immagazzini la frequenza di ogni tono della scala temperata; a tal fine è necessario soltanto specificare il nome dell'array e il numero dell'elemento di ogni nota che si intende usare in un comando SOUND. Col seguente programma si crea un array per ciascun tono della scala temperata, che verrà suonata in tutta la sua successione:

```
DIM pitch(118)
halfstep# = 2#^ (1#/12#)
frequency# = 13.75
FOR i = 1 TO 8
    frequency# = halfstep# * frequency#
NEXT i
FOR i = 4 TO 116
    frequency# = halfstep# * frequency#
    pitch(i) = frequency#
NEXT i
FOR i = 4 TO 116
    SOUND pitch(i), 4, 255
NEXT i
```

Esiste la probabilità che quando si faccia girare il programma non si sentano tutti i toni, poiché alcuni di essi superano le possibilità dell'altoparlante del monitor, e la forma d'onda sinusoidale prefissata che suona le note è difficilmente udibile sui toni bassi. Se assegnerete una forma d'onda a gradino al canale 0 con un comando WAVE all'inizio del programma, dovreste poter sentire le note basse più distintamente.

Un punto interessante di questo programma consiste nel fatto che moltiplicando una frequenza per la radice dodicesima di due (equivalente a 2^ (1/12) nell'Amiga BASIC), la frequenza risultante sarà superiore di un semitono rispetto alla frequenza originaria. La variabile *halfstep#* viene definita come radice dodicesima di 2 nella seconda linea di programma ed è utilizzata nella quinta e ottava linea come fattore per ottenere aumenti di semitoni nella frequenza. Il programma inizia con una frequenza di base di 13.75, un'ottava inferiore al LA_0, per accordare l'intera scala sui 440 cps del LA_4 (13.75 equivale a 440 diviso cinque volte per 2, ed è pertanto cinque ottave inferiore al LA440; ricordate che dimezzare una frequenza significa ridurre il tono di un'ottava). La variabile *frequency#* contiene i valori correnti della frequenza. Le due variabili sono entrambe

a doppia precisione (indicata da "#") e utilizzano costanti anch'esse a doppia precisione per assicurare esattezza di intonazione anche dopo numerose moltiplicazioni.

Il primo ciclo del programma moltiplica *frequency#* per *halfstep#* per creare otto semitoni, ma non li immagazzina nell'array del tono, perché sono tutti al di sotto dei 20 cps e non possono essere usati nel comando SOUND. Il secondo ciclo, che va da 4 a 116, continua a moltiplicare *frequency#* per *halfstep#* al fine di ricavare nuove frequenze di semitoni, partendo dal punto in cui si è concluso il primo ciclo. Il secondo ciclo immagazzina però i semitoni negli elementi dell'array *pitch*, partendo dall'elemento 4, il tono più basso, e terminando con l'elemento 11, la tonalità più alta. La serie parte da 4 poiché il tono più basso contenuto nell'array è MI_0, che è il tono più basso accettato dall'Amiga BASIC. Assegnando a tale tono il numero 4, tutti i DO successivi, all'inizio di ogni nuova ottava, vengono numerati come multipli di 12. In questo modo sarà più facile ricordare gli elementi dell'array di toni, come si potrà vedere in seguito.

L'ultimo ciclo alla fine del programma suona i toni immagazzinati nella serie, partendo da *pitch(4)* per terminare a *pitch(116)*.

Per decidere quale tono dell'array utilizzare in un comando SOUND, moltiplicate il numero dell'ottava del tono desiderato per 12, quindi sommate il risultato con un numero della tabella seguente corrispondente al tono desiderato:

	DO	0	FA♯ o	SOL♭	6
DO♯ o	RE♭	1		SOL	7
	RE	2	SOL♯ o	LA♭	8
RE♯ o	MI♭	3		LA	9
	MI	4	LA♯ o	SI♭	10
	FA	5		SI	11

Figura 9-5.

Notazione musicale tradizionale: posizione e nome delle note.

Ad esempio, LA_4 è il tono dell'elemento numero 57 (12 per 4, più 9). Allo scopo di suonare un LA_4 per un secondo, utilizzerete:

```
SOUND pitch(57), 18.2
```

La vera utilità di un array di toni diventa evidente quando si intende suonare un brano servendosi di comandi SOUND.

ESECUZIONE DI UNA PARTITURA

Qualsiasi componimento musicale che si voglia realizzare sull'Amiga conterrà certo parecchie note. Se si creano le note con un comando SOUND per ciascuna di esse, si rende necessario un programma lungo e tanto tempo libero. Una soluzione ben più semplice consiste nell'utilizzare il comando SOUND in un ciclo che legga i valori dei toni, della durata, del volume e del canale, dai comandi DATA contenenti i dati per la partitura. Le linee del seguente programma compiono esattamente queste funzioni. Suonano un breve componimento (una fioritura, per la precisione) a quattro voci, che inizia dolcemente per poi diventare progressivamente più intenso.

Tali linee non costituiscono un programma completo; affinché operino correttamente, è necessario un array di toni come quello dell'ultimo esempio. Per sentire la fioritura, sostituite le linee del programma seguente con le ultime tre linee dell'esempio di array mostrato in precedenza. Poi fate nuovamente girare il programma.

```
SOUND WAIT
FOR i = 0 TO 3
    SOUND 30, 5, 0, i
NEXT i
SOUND RESUME

Loop:
    FOR i = 0 TO 3
        READ pitch(i), octave(i), duration(i), volume(i)
        IF pitch(i) = -1 GOTO KillTime
        SOUND pitch((12*octave(i))+pitch(i)), 32/duration(i),
        volume(i), i
    NEXT i
    GOTO Loop

KillTime:
    FOR t = 1 TO 4000: NEXT t: END

MusicData:
    DATA 7,4,16,63, 7,3,4,63, 4,3,4,63, 0,3,4,63
    DATA 0,5,16,87, 7,3,4,159, 2,3,4,159, 11,2,4,159
    DATA 11,4,16,111, 7,3,4,255, 4,3,4,255, 0,3,4,255
    DATA 9,4,16,135, 0,0,16,0, 0,0,16,0, 0,0,16,0
    DATA 9,4,16,159, 0,0,16,0, 0,0,16,0, 0,0,16,0
    DATA 9,4,16,183, 0,0,16,0, 0,0,16,0, 0,0,16,0
    DATA 9,4,16,207, 0,0,16,0, 0,0,16,0, 0,0,16,0
    DATA 9,4,16,231, 0,0,16,0, 0,0,16,0, 0,0,16,0
    DATA 9,4,16,255, 0,0,16,0, 0,0,16,0, 0,0,16,0
    DATA -1,-1,-1,-1
```

Il primo ciclo in queste linee, posto a metà fra un comando SOUND WAIT e un comando SOUND RESUME, fa sì che ciascuna delle quattro voci effettui una pausa della medesima durata. I comandi SOUND WAIT e SOUND RESUME sincronizzano le quattro pause in modo che inizino simultaneamente. Durante le pause il programma legge i dati dell'array *pitch* per poi utilizzarli allo scopo di creare note col comando SOUND successivo. Essendo le pause non ancora concluse, tali note vengono bloccate in ogni canale audio finché le pause non sono terminate. Poiché queste ultime terminano contemporaneamente, le note dei comandi SOUND inizieranno tutte assieme, dunque le quattro voci della fioritura saranno sincronizzate.

Questo metodo, basato sull'uso di un ciclo che crea una breve pausa all'inizio di una partitura per sincronizzare le note, invece che sull'uso di comandi SOUND WAIT e SOUND RESUME, è preferibile per spartiti molto lunghi. Questo perché i contenitori per ciascun canale contengono soltanto 12 note alla volta e pertanto si renderebbe necessario l'uso di una serie continua di comandi SOUND WAIT e SOUND RESUME per sincronizzare una lunga partitura. Se si fa girare un programma contenente una serie molto lunga di comandi SOUND WAIT e SOUND RESUME, la musica "singhiozzerà" al termine di ogni contenitore e all'inizio di un altro. Il metodo del ciclo di pause utilizzato nel nostro esempio non è indispensabile per una fioritura così breve; tuttavia sarà molto pratico se si vorranno sostituire i comandi DATA dell'esempio con i propri comandi DATA per un componimento musicale più lungo.

Il ciclo per la sincronizzazione è seguito da un ciclo infinito chiamato *Loop:*. In esso è incluso un ciclo FOR...NEXT che effettua un passaggio per ciascun canale audio (da 0 a 3). Ogni volta che il ciclo FOR...NEXT effettua un passaggio, si serve di un comando READ per leggere i quattro parametri di tono, ottava, durata e volume, che poi immagazzina in quattro variabili. Nella linea successiva, il programma controlla che il tono sia -1, valore usato come segnale al termine dei dati musicali per indicare la fine della partitura. Se il valore è -1, il programma esce dal ciclo e termina.

Se invece non si tratta della fine della partitura, la linea successiva del ciclo FOR...NEXT moltiplica il valore dell'ottava per 12 e vi somma il valore del tono per ottenere il giusto elemento dell'array da utilizzare come frequenza nel comando SOUND. La durata della nota viene stabilita dividendo un certo valore (in questo caso è stato scelto arbitrariamente 32) per il valore della durata letto dal comando DATA. Il volume della

nota è fissato utilizzando il valore del volume letto direttamente dal comando DATA.

Dopo quattro passaggi (uno per ogni canale audio), il ciclo FOR...NEXT si conclude e il ciclo *Loop:* si ripete, facendo girare più volte il ciclo FOR...NEXT, finché questo non incontra il valore -1 nei comandi DATA. Letto quel valore, il programma passa a *KillTime:* e un breve ciclo FOR...NEXT lascia passare del tempo prima della fine del programma, in modo che la finestra List non appaia troppo presto. Se la finestra List appare mentre la musica è ancora in corso, il suono non risulterà fluido.

I dati delle note nei comandi DATA sono organizzati secondo il seguente criterio. Ogni linea DATA contiene quattro note, una per ogni voce, secondo l'ordine 0, 1, 2 e 3. Ogni nota ha quattro valori: il tono, la durata, l'ottava e il volume. Il tono è un numero intero compreso tra 0 e 11, che corrisponde al nome di una nota (LA, FA, MI♭...), come discusso precedentemente nella parte dedicata alle scale dei toni. L'ottava è un numero intero da 0 a 9, corrispondente all'indice del tono indicato da quel numerino che segue sempre il nome della nota.

Per durata si intende un valore che determini la lunghezza della nota: 1 per una semibreve, 2 per una minima, 4 per una semiminima, 8 per una croma, 16 per una biscroma... Questi valori non stabiliscono di per sé la durata delle note; essi vengono semplicemente usati dal comando SOUND per dividere un numero (in questo caso 32) che rappresenta la durata di una nota da 4/4 (semibreve). Cambiando tale numero si viene a modificare il tempo della musica, poiché tutti i valori di durata dei comandi DATA sono divisi per questo numero, allo scopo di determinare la lunghezza di ogni nota. Cambiate ad esempio il 32 con un 64 e fate girare nuovamente il programma; sentirete la fioritura con un tempo dimezzato rispetto all'esecuzione precedente. Sostituendo invece il 32 con un 16, il tempo diventerà doppiamente più veloce.

Il volume rappresenta l'ultimo valore della nota e stabilisce il volume del comando SOUND.

Ogni comando DATA contiene quattro note, una per ciascuna voce. All'interno di ogni comando DATA, i dati di ciascuna nota sono separati con uno spazio dai dati della nota successiva, in modo da rendere facilmente distinguibili le quattro note specificate in ogni linea. La voce 0 è rappresentata dal gruppo che segue immediatamente la parola DATA. È importante notare che il valore della durata della voce 0 in ogni linea è uguale a 16, valore che indica le semicrome. Le voci 1, 2 e 3 suonano note semiminime (come indica il numero 4 per il valore della durata) e richiedono che solo i primi tre comandi DATA immagazzinino i dati relativi alle note. Poiché la voce 0 possiede un numero maggiore di note rispetto alle altre tre

voci, i valori nei restanti comandi DATA per le altre voci diventano 0, 0, 16, 0, e dunque creano una pausa della lunghezza di una semicroma al fine di conservare uno spazio per quelle voci. In questo modo, la voce 0 può continuare la lettura dei suoi dati. Se la durata della voce 0 viene specificata, il programma cercherà di dividere il valore per 0, commettendo un errore di condizione di impossibilità.

Al termine dei comandi DATA vi sono quattro -1, che stanno a indicare la fine della partitura. I -1 sono in numero di quattro poiché il programma legge i dati solo a gruppi di quattro.

Servendosi di questo sistema di comandi READ e DATA per immagazzinare le note, è possibile creare partiture nuove e più lunghe, semplicemente tramite l'uso di dati differenti nei comandi DATA. Se si fa precedere il programma da comandi WAVE per cambiare il timbro dei diversi canali audio, si può assegnare a ciascuna voce un timbro particolare.

SINTESI VOCALE

Il software sistema dell'Amiga comprende libraries e devices che generano la sintesi vocale utilizzando lo stesso hardware e gran parte delle stesse routine software di cui l'Amiga si serve per creare la musica e gli effetti sonori. Il sistema articola parole utilizzando forme d'onda ricavate dalle apposite tavole e riprodotte attraverso i soliti quattro canali audio. Non è necessario definire le forme d'onda, né ricorrere a strani comandi SOUND, poiché l'Amiga BASIC mette a disposizione dell'utente strumenti con varie prestazioni. La funzione TRANSLATE$() e il comando SAY sono utilizzabili per sfruttare direttamente le capacità di creazione di linguaggio dell'Amiga. È dunque possibile aggiungere parti parlate a un programma.

SAY è il comando BASIC che fornisce all'Amiga le istruzioni per parlare e ordina cosa dire. Tale comando richiede che tutte le parole vengano compitate foneticamente, utilizzando un sistema di speciali codici fonetici descritti nell'Appendice H del manuale dell'Amiga BASIC. Se non si desidera compitare tutte le parole foneticamente, è possibile far uso della funzione TRANSLATE$(), che traduce la stringa di una parola nei codici fonetici necessari per il comando SAY, secondo le regole della lingua inglese.

LA FUNZIONE TRANSLATE$()

La funzione TRANSLATE$() traduce le parole inglesi in una stringa di codici fonetici utilizzabili dal comando SAY. La funzione ha la seguente sintassi:

```
TRANSLATE$(''stringa di testo'')
```

La stringa di testo può essere una qualsiasi stringa di caratteri il cui uso è consentito dall'Amiga BASIC, oppure una variabile stringa che rappresenti il testo da esprimere vocalmente. È necessario che la stringa venga introdotta fra parentesi e utilizzando le virgolette, mentre per la variabile stringa sono sufficienti le parentesi.

Essendo una funzione, TRANSLATE$() non può operare da sola. La stringa di codici fonetici ottenuta con questa funzione deve essere assegnata a una variabile stringa utilizzata direttamente dal comando SAY o stampata col comando PRINT. L'uso diretto di TRANSLATE$() con SAY è molto comodo. Ad esempio,

```
SAY TRANSLATE$(''I speak English very well, thank you.'')
```

riproduce vocalmente la frase racchiusa fra virgolette.

Per vedere il codice di fonemi creato da TRANSLATE$(), è possibile servirsi del comando PRINT. Semplicemente,

```
PRINT TRANSLATE$(''I speak English very well, thank you.'')
```

stampa il codice che SAY ha utilizzato, e cioè

```
AY SPIY4K IY3NXGLIHSH VEH1RIY WEH4L, THAE4NXK YUW.
```

La stringa elaborata da TRANSLATE$() si presenta come una serie di lettere maiuscole, poiché il comando SAY richiede espressamente che le lettere compitate foneticamente siano maiuscole. TRANSLATE$() restituisce una stringa di lettere maiuscole, quindi usate tranquillamente le lettere maiuscole quando scrivete qualche frase che volete far verbalizzare al computer.

Se una frase viene stampata sotto forma di fonemi, è possibile modificarla leggermente a propria discrezione e, in

seguito, utilizzarla direttamente con SAY; ciò vi permette di regolare la pronuncia e l'inflessione (vedere Appendice H del manuale dell'Amiga BASIC). Ad esempio, si potrebbe desiderare di porre meno enfasi sulla parola "thank you" (grazie) alla fine della frase. A tal fine, l'operatore può usare un comando PRINT con la funzione TRANSLATE$() per stampare la stringa di fonemi e poi inserirla in un comando SAY cancellando il 4 in THAE4NXK. In questo modo la parola verrà accentata in maniera da toglierle la sua inflessione originaria. Ma attenzione: se i cambiamenti apportati alle stringhe tradotte non si conformano alle istruzioni specificate nell'Appendice H del manuale dell'Amiga BASIC, la stringa potrebbe essere respinta o semplicemente non verbalizzata.

USO DELLA PUNTEGGIATURA

Quando si utilizzano stringhe con TRANSLATE$(), è possibile fare uso dei segni di interpunzione per alterare l'inflessione del testo orale. Tali variazioni vengono effettuate come segue:

- **Punto**: il punto dovrebbe essere usato alla fine di ogni frase. Fa infatti calare il tono della voce per indicare che la frase è finita. Ad esempio:

```
SAY TRANSLATE$(''This is the end.'')
```

- **Punto di domanda**: un punto interrogativo a conclusione di una frase provoca lo stesso effetto di un punto semplice, cioè fa calare il tono della voce. Ad esempio,

```
SAY TRANSLATE$(''This is the end?'')
```

  ha la stessa inflessione dell'enunciato precedente.

- **Virgola**: la virgola interposta fra due parole alza il tono finale della prima parola, ponendo una pausa fra i due termini. Una virgola alla fine della frase ne eleva il tono come se si trattasse di una domanda e può quindi sostituire efficacemente il punto interrogativo.
  Ad esempio:

```
SAY TRANSLATE$(''Well, should I stop now,'')
```

- **Parentesi**: racchiudere un enunciato fra parentesi significa separarlo dal resto della frase, ottenendo un aumento del tono e una breve pausa alla fine dell'enunciazione. Ad esempio:

```
SAY TRANSLATE$(''I think (I really do) that I should stop now.'')
```

- **Trattini**: Il collegamento delle parole tramite trattino fa sì che esse vengano proferite con un'inflessione di gran lunga inferiore rispetto a quella utilizzata per le parole separate da spazi. Ad esempio:

```
SAY TRANSLATE$(''I'm so tired I'm-losing-my-inflection'')
```

COME SI MODIFICA LA COMPITAZIONE PER OTTENERE UNA PRONUNCIA CORRETTA

Talvolta la lingua inglese riesce a mettere in crisi la funzione TRANSLATE$() e dal comando SAY escono pronunce alquanto strane. È il caso, soprattutto, di nomi e termini inglesi che utilizzano una pronuncia straniera. Naturalmente è sempre possibile ricreare le parole con i codici fonetici, però potreste trovare più semplice inserire qualche errore d'ortografia, che si può rivelare di grande aiuto per la funzione TRANSLATE$().

TRANSLATE$(), ad esempio, incontra difficoltà notevoli con il nome Michael, che pronuncia più o meno "Mitch ay ell". Una semplice revisione ortografica risolverà il problema. Per sentire la differenza, provate i due seguenti comandi SAY:

```
SAY TRANSLATE$(''Michael'')
SAY TRANSLATE$(''Mikul'')
```

Con un po' di pratica capirete subito come TRANSLATE$() pronuncia l'inglese e supererete ogni ostacolo. Nel complesso, però, le capacità della funzione suscitano sempre grande interesse e stupore. Chi conosce l'inglese potrà senza dubbio capire di cosa è capace una funzione che riesce a pronunciare correttamente la "o" della parola "women".

IL COMANDO SAY

Il comando SAY viene utilizzato per ordinare all'Amiga cosa dire e come dirlo. La sintassi di SAY è la seguente:

```
SAY stringa di fonemi, nome dell'array di specificazione
```

La stringa di fonemi è costituita da lettere maiuscole e numeri che compitano foneticamente le parole utilizzando i codici descritti nell'Appendice H del manuale dell'Amiga BASIC. Il nome dell'array di specificazione è opzionale; è il nome di un array di numeri interi che immagazzina nove differenti valori, i quali indicano il modo in cui il comando SAY si esprimerà. Se non viene specificato alcun nome, il comando SAY verbalizzerà la stringa di fonemi utilizzando la voce maschile prefissata.

I CODICI FONETICI

L'inglese è una lingua difficile da compitare e pronunciare. La stessa lettera, in due parole diverse, può essere pronunciata in due modi differenti, come ad esempio la prima "c" di "cycle", pronunciata come una "s", e la "c" di "call", che invece è dura. Per ordinare con precisione a SAY la pronuncia desiderata, è necessario spezzare le parole in suoni ben distinti, detti fonemi, e utilizzare gli appositi codici dell'Amiga per compitare i fonemi ricavati. Il sistema più semplice di svolgere tale operazione consiste naturalmente nel far svolgere tutto il lavoro alla funzione TRANSLATE$(). Tuttavia, poiché TRANSLATE$() non interpreta correttamente tutte le parole e lavora solo con l'inglese, è probabile che si verifichi la necessità di ricorrere alla compitazione fonetica, al fine di ottenere che SAY si esprima con la massima precisione possibile.

L'Appendice H del manuale dell'Amiga BASIC spiega dettagliatamente come compitare una parola con i codici fonetici. Sebbene sia possibile utilizzare la funzione TRANSLATE$(), se deciderete di imparare i codici non sprecherete certo il vostro tempo: in questo modo riuscirete a far parlare il computer in maniera pressoché perfetta. I codici sono molto utili per specificare accenti ben precisi, pause ed enfasi di una frase, oppure per far parlare il computer in lingue diverse dall'inglese. Ad esempio, il seguente comando SAY chiede "Wie geht es Ihnen?", che in tedesco significa "Come sta?".

```
SAY ''IY2 GEY7T IX2S IY3NIXN.''
```

L'ARRAY DI SPECIFICAZIONE

Se non si vuole che SAY utilizzi la voce di default (maschile), la qualità della sua espressione orale può essere mutata fissando nove differenti aspetti del discorso. Invece di inserire una lunga lista di nove specificazioni alla fine di ogni comando

SAY, è possibile immagazzinare tali indicazioni in un breve array nel quale SAY andrà a leggere i valori tutte le volte che nel comando viene specificato il nome. L'array deve essere intero (indicato quindi da un simbolo % alla fine del nome della variabile) e contenere nove elementi. I valori di specificazione memorizzati nei nove elementi (0-8) controllano il tono, l'inflessione, la velocità di espressione, il volume della voce e altri aspetti dell'oralità. La seguente tabella mostra la caratteristica controllata da ciascun elemento dell'array:

Numero elemento	Aspetto controllato
0	Tonalità di base della voce
1	Inflessione
2	Velocità
3	Tipo di voce
4	Frequenza di campionamento
5	Volume
6	Assegnazione del canale
7	Sincronizzazione delle frasi
8	Esecuzione di più comandi SAY

Regolazione della tonalità di base

L'elemento numero 0 dell'array deve essere un numero intero compreso fra 65 e 320. Esso dà la frequenza di base del tono della voce espresso in cps. Il valore di default è 110, corrispondente a un LA della seconda ottava. Quando SAY verbalizza la stringa oralmente, aggiunge alle parole una certa inflessione, e pertanto il tono di base è solo il tono centrale del discorso. L'inflessione alza e abbassa il tono rispetto al suo valore di partenza.

Selezione dell'inflessione o monotòno

L'elemento numero 1 dell'array deve essere 0 o 1, ed è utilizzato per specificare se le frasi abbiano una certa inflessione o siano semplicemente monotòni senza cadenza. 0 indica una pronuncia con inflessione, mentre 1 specifica l'espressione piatta e monotòna.

L'inflessione viene spesso usata per dare vitalità alle enunciazioni dell'Amiga. Se si desidera che SAY mantenga un tono di base costante, conviene usare il monotòno; quest'ultimo imita perfettamente la voce metallica dei robot.

Regolazione della velocità

L'elemento numero 2 dell'array deve essere un numero intero compreso fra 40 e 400; specifica il tasso di uscita delle parole al minuto. Il tasso di default è pari a 150 parole al minuto, corrispondente alla normale velocità del linguaggio. Il tasso di 400 è ottimo per gli scioglilingua; diminuendo il tasso al di sotto di 100, si ottiene una pronuncia lenta e biascicata.

Selezione del tipo di voce

L'elemento numero 3 dell'array determina il tipo di voce usato da SAY. Il valore può essere 0 o 1. Il valore di default, 0, seleziona una voce maschile, mentre 1 utilizza una voce femminile. Cambiando il genere di voce, si modifica il modo in cui SAY crea i suoi fonemi, e quindi si trasforma il suono caratteristico del suo linguaggio in un suono più simile a quello emesso dalle cavità orali di un uomo o di una donna. La voce maschile è quella che si adatta meglio al tono di base di 110 cps. Se si seleziona la voce femminile, è necessario convertire anche la tonalità di base in 220 cps o più (un aumento di un'ottava), perché la voce della maggior parte delle donne è di un ottava superiore rispetto a quella degli uomini.

Regolazione della frequenza di campionamento

L'elemento numero 4 specifica la frequenza di campionamento della voce di SAY in cicli al secondo. La frequenza può essere indicata da qualsiasi numero intero compreso fra 5000 e 28000. La frequenza di default è 22000. La frequenza di campionamento stabilisce la velocità con cui l'Amiga riproduce la tavola di forme d'onda utilizzata per creare una voce. Si può considerare questo valore come l'equivalente audio della risoluzione grafica. In generale, quanto più elevata è la frequenza, tanto più fluida è la voce (sebbene una frequenza troppo elevata produca una voce stridula); quanto minore è la frequenza, tanto più la voce è cupa e bassa.

La frequenza di campionamento e il tono di base sono interdipendenti. Come spiegato nel settimo capitolo, ogni volta che si aumenta la velocità di riproduzione di una tavola di forme d'onda, si eleva il tono del suono emesso. Allo stesso modo, aumentando la frequenza di campionamento vocale, si eleva il tono della voce. Diminuendola, invece, il tono scende. Prima di cambiare la frequenza di campionamento per modificare il carattere della voce, si dovrebbe ricercare la qualità generale della voce desiderata utilizzando il tono prefissato di 22000 e variando le altre caratteristiche della voce. Una volta trovata la qualità della voce, ristabilite il tono di base per ricercare il nuovo tono desiderato.

Regolazione del volume

L'elemento numero 5 regola il volume di SAY. Può essere qualsiasi numero intero compreso fra 0 e 64. 0 indica la mancanza di volume (silenzio). 64 indica il volume massimo, mentre i valori intermedi sono fissati dai numeri posti fra i due estremi. Il volume di default è 64.

Assegnazione del canale

L'elemento numero 6 dell'array offre 12 opzioni differenti per il canale o canali audio utilizzati da SAY per parlare. I canali vengono specificati utilizzando un numero intero compreso fra 0 e 11 per designare una delle assegnazioni riportate nella tabella seguente:

0	=	Canale 0
1	=	Canale 1
2	=	Canale 2
3	=	Canale 3
4	=	Canali 0 e 1
5	=	Canali 0 e 2
6	=	Canali 3 e 1
7	=	Canali 3 e 2
8	=	Qualsiasi canale disponibile che esca dalla porta audio di sinistra
9	=	Qualsiasi canale disponibile che esca dalla porta audio di destra
10	=	Qualsiasi coppia disponibile di canali che escano da entrambe le porte audio
11	=	Qualsiasi canale singolo disponibile

Le opzioni da 0 a 3 funzionano allo stesso modo dell'assegnazione del canale nel comando SOUND. Le opzioni da 4 a 7 sono coppie di canali che utilizzano le porte audio sia di destra che di sinistra, in modo da consentire l'ascolto della voce da entrambe. Le opzioni 8 e 9 consentono di specificare la porta audio destra o quella sinistra, lasciando però al BASIC la scelta del canale. È un'opportunità molto vantaggiosa se si utilizza SAY mentre i comandi SOUND e BEEP stanno producendo altri suoni sui canali audio; SAY può infatti utilizzare qualsiasi canale rimasto libero.

L'opzione 10, relativa all'assegnazione del canale, specifica qualunque coppia di canali lasciati liberi da altri comandi sonori, che si servono di entrambe le porte audio. In altre parole, SAY utilizzerà qualsiasi coppia di canali definita dalle opzioni 4, 5, 6 e 7. Infine, l'opzione 11 seleziona qualsiasi canale disponibile non utilizzato da altri comandi. Si tratta forse della

scelta più sicura, poiché è molto più probabile trovare libero un singolo canale audio che non una coppia. Tuttavia, se si opta per la scelta 11 e l'Amiga è collegato a un impianto stereo con due casse, non sarà possibile prevedere da quale cassa proverrà la voce.

Selezione del discorso sincrono e asincrono

L'elemento numero 7 dell'array può essere 0 o 1. Il valore di default, 0, indica il discorso sincrono, mentre 1 indica quello asincrono. Se si seleziona il discorso sincrono, il BASIC attenderà che ogni comando SAY abbia finito di parlare e poi passerà al comando successivo. Se invece si sceglie il discorso asincrono, il BASIC eseguirà un comando SAY, dopodiché farà girare il comando successivo non appena il primo avrà completato la sua funzione.

Il linguaggio asincrono velocizza i tempi del programma, poiché quest'ultimo può operare con alcuni comandi e contemporaneamente parlare. Tuttavia, altri comandi per la generazione di suoni (come SOUND e BEEP), che seguono un comando SAY asincrono, possono ingarbugliare i discorsi di SAY, iniziando a operare prima che SAY abbia finito di parlare. Il discorso sincrono rende impossibile l'insorgenza di simili problemi.

Esecuzione di più comandi SAY

L'elemento numero 8, l'ultimo dell'array, specifica il modo in cui l'Amiga BASIC esegue i comandi SAY se essi utilizzano il discorso asincrono (ovvero se l'elemento 7 è regolato sullo 0). Esistono tre scelte:

Valore	Significato
0	Se il BASIC incontra un secondo comando SAY prima che il primo abbia completato la sua funzione, attenderà che il primo comando abbia finito di operare prima di iniziare a eseguire il secondo. Lo 0 è il modo di default.
1	Se il BASIC incontra un secondo comando SAY prima che il primo abbia finito di operare, interrompe completamente l'emissione di parole in modo che il primo comando venga troncato a metà del discorso e il secondo non agisca affatto.
2	Se il BASIC incontra un secondo comando SAY prima che il primo abbia terminato la sua funzione, interrompe il primo comando a metà del discorso e fa partire immediatamente il messaggio del secondo comando.

Lo scopo di queste opzioni è di evitare confusioni, impedendo a più voci di parlare contemporaneamente.

ESEMPI DI COMANDO SAY

L'uso di un array di specificazione allo scopo di alterare la qualità dell'espressione orale di SAY non presenta difficoltà. Non è necessario dimensionare l'array, poiché contiene meno di 11 elementi, ed è possibile usare un comando READ in un ciclo FOR...NEXT, al fine di leggere i valori di ciascun elemento da un comando DATA. Ad esempio, il breve programma che segue è predisposto in modo che il comando SAY utilizzi una voce femminile suadente. Il programma immagazzina le specificazioni per il comando SAY nell'array *voice%* successivamente utilizzato nel comando SAY che si produce nel seguente scioglilingua: "Peter Piper picked a peck of pickled peppers" (una sorta di "Trentatrè trentini..." in lingua inglese).

```
FOR i = 0 TO 8
    READ voice%(i)
NEXT i
x$ = ''Peter Piper picked a peck of pickled peppers.''
SAY TRANSLATE$(x$), voice%
DATA 250, 0, 400, 1, 24000, 64, 10, 0, 0
```

Quando si fa girare il programma, si nota che esso compie una breve pausa per caricare qualcosa dal dischetto del BASIC. Ciò è dovuto al fatto che il BASIC deve aprire il dispositivo vocale quando incontra per la prima volta un comando SAY. Qualsiasi comando SAY successivo svolgerà la sua funzione immediatamente; il BASIC deve aprire il dispositivo vocale una volta soltanto, ma comunque tutte le volte che si fa girare il programma.

Se si desidera ottenere voci di qualità differenti, è possibile modificare i valori nel comando DATA e far girare nuovamente il programma.

Il programma che segue legge quattro diversi array di specificazione e ne utilizza i valori per creare quattro differenti voci. La prima voce femminile è ottenuta elevando il tono di base da 110 a 220 cps. La seconda voce è simile alla voce laconica di un robot; è creata abbassando il tono di base a 70 cps, selezionando il monotòno e fissando la velocità a 80 parole

al minuto. La terza voce ha un suono nasale; è creata aumentando il tono di base a 320 cps e diminuendo la frequenza di campionamento a 1000 cps. L'ultima voce è assai ricca e profonda ma molto veloce; il tono di base viene abbassato a 65, la velocità è aumentata a 350 parole al minuto e la frequenza di campionamento è elevata a 28000.

```
FOR i = 0 TO 7
    READ voice1%(i), voice2%(i), voice3%(i), voice4%(i)
NEXT i
DATA 220, 70, 320, 65
DATA 0, 1, 0, 0
DATA 150, 80, 110, 350
DATA 1, 0, 0, 1
DATA 22200, 22200, 10000, 28000
DATA 64, 64, 64, 64
DATA 10, 10, 10, 10
DATA 1, 1, 1, 1
SAY TRANSLATE$(''This is voice 1 speaking to you now.''), voice1%
PRINT ''This is voice 1 speaking to you now.''
SAY TRANSLATE$(''This is voice 2 speaking to you now.''), voice2%
PRINT ''This is voice 2 speaking to you now.''
SAY TRANSLATE$(''This is voice 3 speaking to you now.''), voice3%
PRINT ''This is voice 3 speaking to you now.''
SAY TRANSLATE$(''This is voice 4 speaking to you now.''), voice4%
PRINT ''This is voice 4 speaking to you now.''
FOR x =  1 TO 3000: NEXT x
```

Quando fate girare il programma, si nota che, sebbene i comandi PRINT eseguano i comandi SAY che descrivono, le frasi vengono stampate sullo schermo nello stesso istante in cui il computer inizia a parlare. Ciò è dovuto al fatto che tutte e quattro le voci sono regolate sul discorso asincrono e quindi il BASIC esegue immediatamente il comando PRINT non appena il comando SAY attacca discorso. Il ciclo FOR...NEXT alla fine del programma impedisce di vedere la finestra List prima della completa esecuzione del quarto comando SAY.

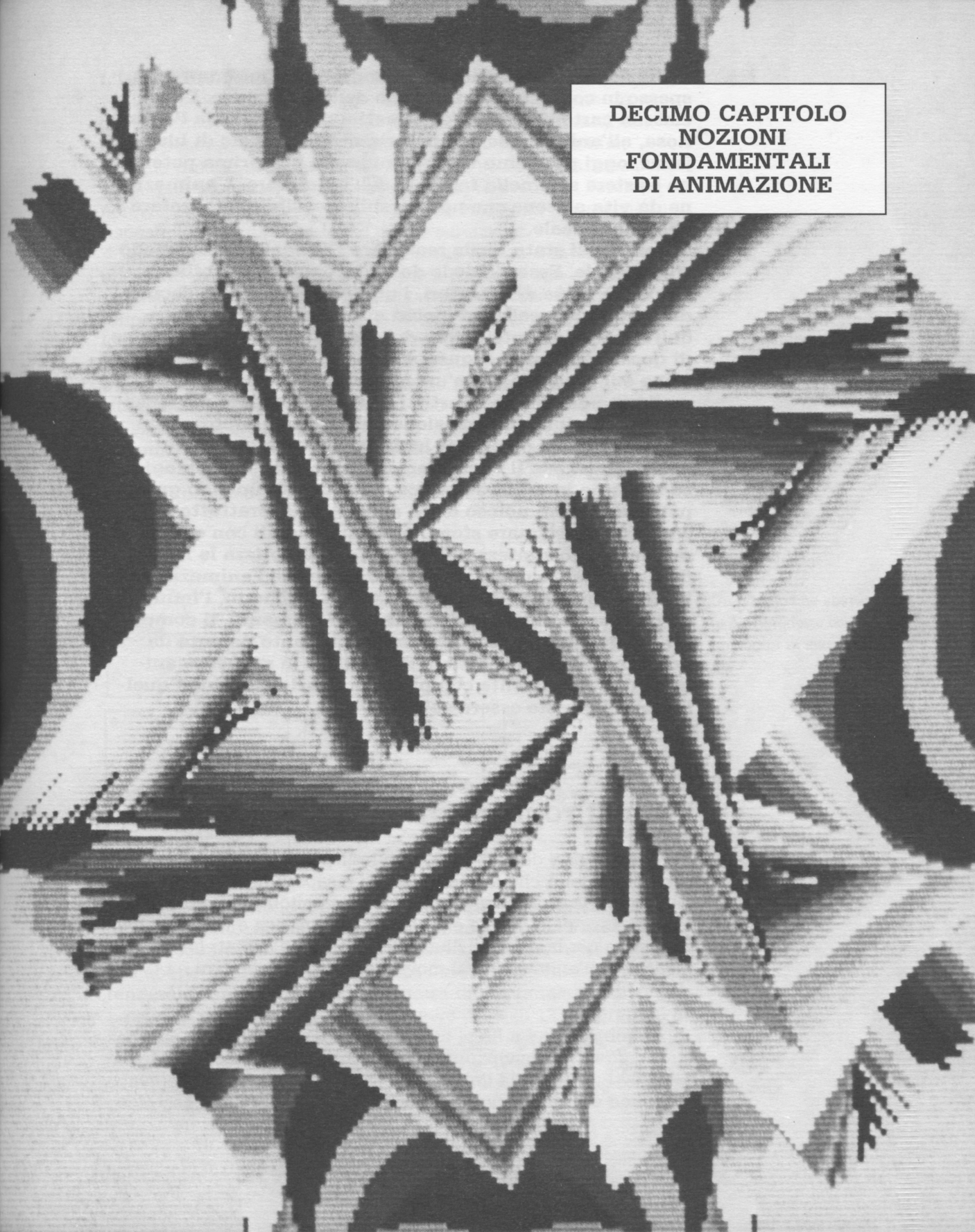
Se cambiate il penultimo comando DATA in una serie di quattro zeri, il comando PRINT attenderà che tutte le voci abbiano finito di parlare, dato che in questo caso i comandi SAY fanno uso del discorso sincrono. Selezionando il discorso sincrono, il ciclo FOR...NEXT alla fine del programma diventa superfluo.

Dopo questa carrellata sui comandi sonori dell'Amiga BASIC, sarete certamente in grado di utilizzarli per creazioni personali. BEEP e SOUND riprodurranno gli effetti sonori e le note musicali, WAVE si occuperà dei timbri, SAY e TRANSLATE$() cureranno le parti parlate. Potete sincronizzare le vostre intricate sequenze sonore con i comandi SOUND WAIT e SOUND RESUME. Utilizzando tutti i comandi esaminati, potete integrare i vostri programmi con colonne sonore, commenti e altri effetti audio. E se vi è piaciuta la sezione dedicata al suono, sarete quanto meno entusiasti della sezione successiva di questo libro, interamente dedicata all'animazione.

SEZIONE 4

Animazioni

In questa sezione parleremo delle possibilità di animazione offerte dall'Amiga.
Il decimo capitolo tratta dei concetti fondamentali dell'animazione computerizzata.
L'undicesimo capitolo spiega come usare Deluxe Video per creare animazioni e presenta altri prodotti hardware e software che vi faciliterranno nella realizzazione delle idee più originali. Gli ultimi due capitoli della sezione (dodicesimo e tredicesimo) illustrano i comandi dell'Amiga BASIC riservati all'animazione, spiegando come combinarli per animare le figure nei vostri programmi BASIC.



DECIMO CAPITOLO
NOZIONI FONDAMENTALI DI ANIMAZIONE

Grazie alla diffusione del televisore, siamo sempre più spesso in contatto con il mondo dell'animazione. Dai classici cartoni animati disegnati a mano come la Pantera Rosa, all'animazione computerizzata di un paio di bluejeans, oggi possiamo vedere immagini che prima potevano esistere solo nella fantasia dell'animatore. L'animazione dà vita a scene che non sarebbe possibile fotografare nel mondo reale.

Non è mai stato facile realizzare l'animazione nel modo tradizionale. Essa richiede da un lato notevole abilità e sforzo da parte dei creatori, i quali devono creare decine di disegni per realizzare pochi minuti di animazione, e dall'altro un attento lavoro dei tecnici per la trasposizione di questi disegni in sequenze filmate. Raramente l'animazione è il prodotto di un'unica persona. In genere è realizzata da équipe di artisti, tutti egualmente responsabili e partecipi della creazione del prodotto finito.

L'avvento del computer ha influenzato anche l'arte dell'animazione. Il computer può infatti creare immagini su schermo in modo semplice e veloce, eliminando gran parte del lavoro noioso degli animatori, soprattutto il continuo ridisegnare sfondi e oggetti mobili con estrema pazienza e accuratezza. Il computer memorizza le immagini, e pertanto non è necessario trasporre l'animazione in filmato. Una volta immagazzinata su dischetto, l'immagine può essere riprodotta in qualsiasi momento. Il computer è dunque un mezzo di animazione tanto potente da poter trasformare le visioni dell'artista in sequenze animate di sorprendente complessità. Grazie all'Amiga quell'artista potreste essere voi.

NOZIONI FONDAMENTALI DI ANIMAZIONE

Per poter fare dell'animazione è indispensabile conoscere i diversi elementi che la costituiscono e saperli usare nel modo più appropriato. Dopo aver chiarito nei capitoli precedenti l'importanza della luce e del colore nella grafica e della qualità del suono per la musica, illustreremo ora due aspetti fondamentali dell'animazione: il movimento e la prospettiva.

L'ILLUSIONE DEL MOVIMENTO

L'animazione è un'illusione. Quello che appare come un continuo movimento di oggetti sullo schermo è in realtà una rapida successione di immagini fisse. Guardando un film, l'occhio umano recepisce 25 immagini al secondo; l'animazione video può presentare fino a 60 immagini al secondo. Perché non riusciamo a distinguere le singole immagini? Due sono le risposte: continuità e velocità di presentazione.

La mente umana ha una propensione innata a interpretare una serie di immagini come un'unica immagine in movimento, purché le immagini proposte siano simili e presentino solo lievi differenze fra una e l'altra. È sufficiente che ci sia continuità logica fra un'immagine e l'altra. La Figura 10-1 presenta sei immagini di un uccello che sbatte le ali. Se vi venissero presentate queste immagini in rapida successione, credereste sicuramente di vedere un uccello che muove realmente le ali. Lo stesso discorso vale anche se le immagini si succedono lentamente. Una delle caratteristiche fondamentali della mente umana consiste proprio nella facilità con cui collega elementi separati per creare un insieme sensato e logico.

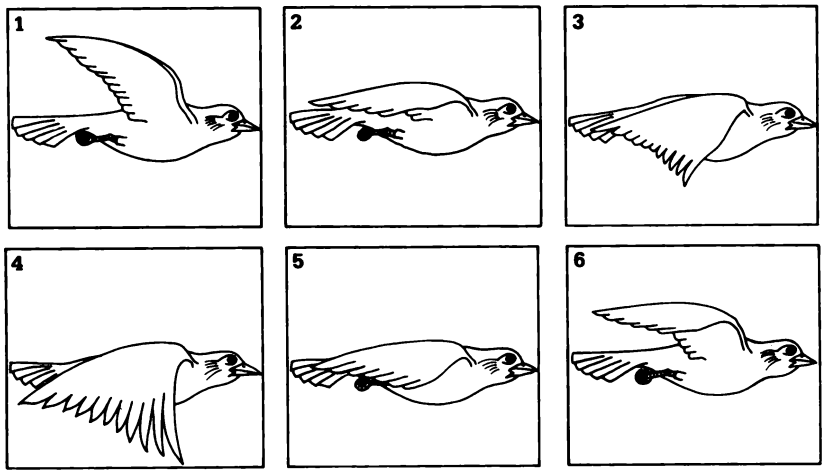


Figura 10-1.

Un uccello che sbatte le ali riprodotto con sei immagini in successione.

Una presentazione rapida delle immagini facilita l'illusione del movimento. Se mostrate le sei immagini della Figura 10-1 a una velocità di sei immagini al secondo, vedrete un uccello che sbatte le ali, ma il movimento risulterà convulso; perciò vi renderete conto del fatto che si tratta di più immagini diverse. Le stesse immagini mostrate a una velocità di 12 fotogrammi al secondo (in modo che l'uccello sbatta le ali a velocità doppia) creeranno un moto più fluido e continuo, quasi per niente convulso. A una velocità di 25 fotogrammi al secondo, i vostri occhi percepiranno un moto uniforme. Ovviamente, se non ci

fosse continuità fra le immagini, nemmeno 25 fotogrammi al secondo riuscirebbero a dare l'illusione del movimento. Per esempio, mostrando diapositive completamente diverse a un ritmo di 25 al secondo non si otterrebbe l'illusione del movimento ma solo un'accozzaglia di immagini: il tutto per mancanza di continuità fra le immagini stesse.

Se un secondo di animazione è dato dal susseguirsi di 25 immagini differenti (dette appunto fotogrammi), se ne deduce che un breve spot pubblicitario di un minuto è costituito da ben 1500 fotogrammi diversi. E creare tutti questi fotogrammi a mano o anche con un computer non è cosa da poco. Con l'aiuto del computer è però possibile evitare di disegnare 1500 fotogrammi distinti e completi.

Uno dei tanti metodi consiste nel creare ogni nuova immagine di una stessa sequenza semplicemente copiando l'immagine precedente e apportando le variazioni necessarie a quelle parti della figura che devono risultare in movimento. Dato che gli elementi fissi si mantengono costanti in ogni fotogramma, è possibile separare gli elementi mobili da quelli fissi e limitarsi a ridisegnare i primi.

Il modo più semplice per separare le due categorie di elementi consiste nel suddividerli in sfondo e oggetti animati. Lo sfondo può essere un muro, una montagna o una foresta, elementi che non si muovono e fungono da retroscena per l'azione degli oggetti animati. Questi possono essere persone, animali, aerei o qualsiasi altra cosa vogliate animare.

Lo sfondo

Lo sfondo è un po' lo scenario in cui si muove l'animazione. Generalmente lo si disegna una volta sola per tutta la sequenza, per cui vale la pena di curarlo nei minimi particolari, anche se la cosa vi potrà costare un po' di tempo.

Qualche volta lo sfondo stesso si muove, in genere per indicare un cambiamento del punto di osservazione. Provate a mettere una mano davanti al viso e guardate le dita, osservando però lo sfondo dietro la vostra mano. Ora girate la testa e la mano lentamente verso destra, continuando a fissare le vostre dita. Rispetto a queste vi sembrerà che lo sfondo si muova verso sinistra. Spostando lentamente lo sfondo verso destra o verso sinistra, verso l'alto o verso il basso, o anche diagonalmente, modificherete la prospettiva della vostra animazione. Nell'animazione tradizionale questa operazione è detta panoramica; nell'animazione computerizzata è chiamata invece "scroll".

Lo sfondo può cambiare anche quando il punto di osservazione si avvicina o si allontana dallo sfondo stesso. Gli oggetti che formano lo sfondo ingrandiscono a mano a mano che l'osservatore immaginario si avvicina, mentre si rimpicciolisco-

no quando l'osservatore si allontana. Ingrandendo cioè lo sfondo, si dà la sensazione che l'osservatore avanzi, mentre restringendolo sembrerà che l'osservatore indietreggi.

Tradizionalmente, gli animatori creano lo sfondo dipingendolo e disponendolo su un apposito sostegno, costituito da una superficie piana simile a un tavolo da disegno e dotata di una cinepresa che fotografa un fotogramma alla volta. L'animatore può usare tale sostegno per muovere lo sfondo, o spostando leggermente il disegno di fotogramma in fotogramma (panoramica dello sfondo), oppure avvicinando o allontanando la cinepresa rispetto al disegno fisso (avvicinando o allontanando il punto di osservazione). Con questa operazione si perde del tempo, in quanto, a ogni sequenza di movimento dello sfondo, l'animatore deve continuare a spostare il disegno di sfondo o la cinepresa, impressionare un fotogramma, per poi spostare di nuovo disegno o cinepresa.

La stessa operazione svolta su un computer risulta molto più semplice. Lo sfondo viene creato con il software grafico e immagazzinato nella memoria del computer. Memorizzato il disegno, l'animatore si limiterà a dare istruzioni al computer perché faccia una panoramica dello sfondo facendolo scorrere nella direzione desiderata, o perché modifichi la prospettiva stringendo o allargando l'immagine attraverso una variazione dei dati dell'immagine stessa.

Oggetti animati

Sistemato lo sfondo, l'animatore deve creare degli oggetti che si muovano su di esso. Ciascuno di questi oggetti può muoversi con due diversi tipi di moto: esterno e interno. Il movimento esterno è quello dell'intero oggetto rispetto allo sfondo. Il movimento interno è quello delle singole parti dell'oggetto. Prendete come esempio un uomo che attraversa la strada. Il suo movimento esterno è costituito dal corpo che va da un lato all'altro della strada, mentre il movimento interno è dato dalle gambe che si muovono avanti e indietro, dalle braccia che ciondolano, dal busto che va su e giù e da un largo sorriso che gli illumina il viso.

Nell'animazione tradizionale, l'animatore crea oggetti mobili disegnandoli o dipingendoli su fogli trasparenti di acetato (chiamati "gel" o fogli di carta da lucido). Questi fogli lucidi vengono posti sul disegno dello sfondo pronto per essere filmato. Dato che i fogli lucidi sono completamente trasparenti fatta eccezione per le zone coperte dagli oggetti disegnati, la cinepresa fotografa i lucidi e lo sfondo assieme, realizzando un fotogramma che sembra essere un unico disegno completo, anche se in realtà sul dipinto di sfondo vi sono più fogli gel.

Per creare il movimento interno, l'animatore tradizionale disegna una serie di fogli lucidi, in ciascuno dei quali l'oggetto animato si trova in una posizione leggermente differente. Fotografando i fogli in successione sul disegno di sfondo, si può creare un'animazione con movimento interno. L'animatore che usa il computer si comporta in modo analogo: crea una serie di immagini, la immagazzina nella memoria del computer ed è poi il computer stesso a visualizzare le immagini in sequenza sullo schermo. Né il metodo tradizionale, né quello computerizzato per la creazione di movimento interno sono semplici. In ambedue i casi si devono disegnare molte immagini, cosa che richiede parecchio tempo.

La creazione di movimento interno può essere però semplificata grazie alla natura ciclica del moto. Per esempio, gli uccelli sbattono le ali verso il basso, poi verso l'alto, poi ancora verso il basso e di nuovo verso l'alto. Tralasciando alcune variazioni di minore importanza che si possono riscontrare tra un colpo d'ala e l'altro, è possibile disegnare una sequenza d'immagini che animi un solo colpo d'ala e continuare poi a ripeterla per ogni movimento dell'uccello. Questa operazione è detta cycling (rotazione). Anziché creare centinaia di immagini per ottenere mezzo minuto di volo, potete creare solo poche immagini e ripeterle più volte. La sequenza di cycling non è all'altezza di una sequenza le cui immagini siano state disegnate individualmente, ma permette comunque di ottenere un buon movimento interno con uno sforzo minimo.

Creata la serie di immagini che animano internamente l'oggetto, è necessario assicurarsi che il movimento interno concordi con quello esterno. Per esempio, se riproducete un uomo che cammina muovendo gambe e braccia, ma non introducete alcun tipo di movimento esterno (o non fate scorrere lo schermo dietro all'uomo), sembrerà che il poverino scivoli sul ghiaccio senza avanzare. E viceversa, se il movimento esterno è esagerato, sembrerà che l'uomo indossi gli stivali delle sette leghe, che lo fanno volare anche col più piccolo dei passi.

INTRODUZIONE DELLA VISIONE IN PROSPETTIVA

Quando è stata completata nei minimi particolari, l'animazione viene visualizzata da mezzi di riproduzione bidimensionali, come il monitor del computer o lo schermo cinematografico. È però possibile utilizzare diverse tecniche di animazione e di disegno che permettono di trascendere la bidimensionalità dello schermo e di rappresentare le immagini in una qualsiasi delle tre diverse prospettive: a due dimensioni (2-D), a tre dimensioni (3-D) e a due dimensioni e mezzo (2,5-D).

Prospettiva a due dimensioni

La prospettiva a due dimensioni parte dal presupposto che non vi sia profondità nell'immagine e che gli oggetti si muovano orizzontalmente e verticalmente, ma non da e verso lo spettatore. La prospettiva bidimensionale si addice in particolare a quel tipo di rappresentazione che vuole essere una rappresentazione simbolica della realtà, pur non avendo la pretesa di apparire vera.

Un buon esempio di questo tipo di animazione è dato dall'immagine delle automobili che si muovono sulla cartina stradale, illustrate nella Figura 10-2. Il loro movimento potrà essere solo orizzontale, verticale o diagonale: tutti movimenti bidimensionali, appunto.

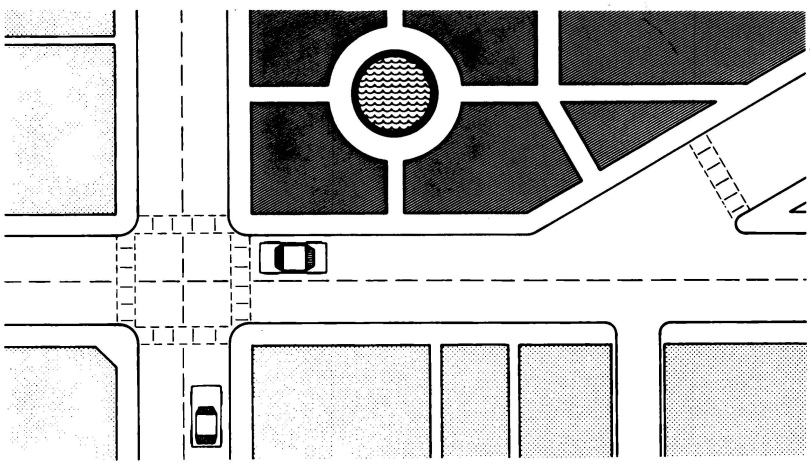


Figura 10-2.

Automobili su una piantina stradale: una semplice animazione bidimensionale.

Un'importante caratteristica dell'animazione a 2-D è il fatto che un oggetto in movimento non può passare davanti o dietro agli altri; gli oggetti si spostano infatti su un piano bidimensionale e pertanto, se si avvicinano troppo, finiscono solo con lo scontrarsi. Le due automobili dell'esempio non possono sovrapporsi, una deve girare intorno all'altra, altrimenti si scontrano.

L'animazione bidimensionale non è molto realistica ma presenta dei vantaggi rispetto alle altre prospettive. Innanzitutto è più semplice da gestire, perché è sufficiente tener conto della posizione dell'oggetto inserito in un piano a due dimensioni, completamente privo di profondità. Inoltre una semplice rappresentazione simbolica può mostrare con maggiore chiarezza gli elementi più importanti della sequenza animata. Se le strade della piantina fossero state disegnate come appaiono effettivamente dall'alto, con rilievi, avvallamenti e superficie asfaltata, l'animazione sarebbe stata più realistica ma avrebbe in parte ostacolato l'osservazione delle macchine in movimento.

Prospettiva a tre dimensioni

La prospettiva a tre dimensioni conferisce senso di profondità e maggior realismo all'immagine sullo schermo. Nell'animazione a 3-D gli oggetti si muovono da e verso lo spettatore, oltre che orizzontalmente e verticalmente. La terza dimensione, la profondità, è essenzialmente un'illusione realizzata su schermo bidimensionale con diverse tecniche di animazione.

La più comune di queste tecniche è quella dell'orizzonte artificiale, un'invisibile linea di orizzonte usata dagli animatori come punto di riferimento. Se non siete pratici di orizzonti artificiali, pensate a un orizzonte vero. Immaginate di trovarvi di fronte a un'immensa distesa d'acqua: la linea lungo la quale il mare sembra incontrare il cielo costituisce l'orizzonte. Le barche che galleggiano sull'acqua vicino alla riva sembrano al di sotto dell'orizzonte. A mano a mano che si allontanano da voi, le barche sembrano salire sempre più in alto, verso l'orizzonte, diventando sempre più piccole fino a scomparire. Allo stesso modo un aeroplano che vola sopra di voi, verso l'orizzonte, scende apparentemente verso quest'ultimo con moto costante e si fa sempre più piccolo fino a sparire.

La Figura 10-3 rappresenta due imbarcazioni dirette verso l'orizzonte. Il natante situato nell'angolo inferiore sinistro, essendo più grande dell'altro e più lontano dall'orizzonte, appare più vicino allo spettatore.

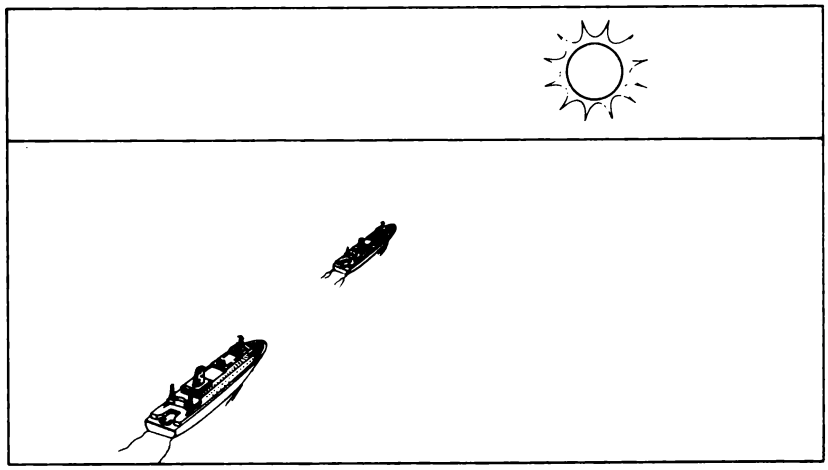


Figura 10-3.

Due barche dirette verso l'orizzonte.

Un orizzonte artificiale funziona allo stesso modo. In genere attraversa il centro dello schermo e può essere invisibile perché nascosto da edifici, alberi, montagne o altro. Anche se è invisibile, la sua posizione può essere usata come punto di riferimento per dare l'illusione che gli oggetti si avvicinino o si allontanino dallo spettatore. Spostando verso l'orizzonte artifi-

ciale un oggetto e contemporaneamente rimpicciolendolo, si dà l'illusione che esso si stia allontanando dall'osservatore. È possibile rimpicciolire l'oggetto fino a farlo scomparire, e sembrerà così che l'oggetto si sia allontanato troppo per poter essere visto. Il punto in cui l'oggetto sparisce è detto punto di fuga. Al contrario, portando l'oggetto lontano dall'orizzonte e contemporaneamente ingrandendolo, si darà l'impressione che l'oggetto si muova verso lo spettatore. Più velocemente l'oggetto si rimpicciolisce o si ingrandisce, più veloce apparirà il suo moto da e verso lo spettatore. È possibile combinare il mutamento di dimensioni con il movimento orizzontale e verticale; in questo modo l'oggetto potrà muoversi in qualsiasi direzione prevista dalla prospettiva tridimensionale dell'animazione.

Per ottenere un maggiore senso di profondità è anche possibile cambiare il colore dell'oggetto mentre si avvicina o si allontana. In genere il colore degli oggetti più lontani è più tenue e sbiadito rispetto a quello degli oggetti più vicini. Affievolendo il colore di un oggetto che si allontana si accrescerà il senso di profondità. Per sviluppare la prospettiva tridimensionale può essere utile anche diminuire il movimento interno di un oggetto a mano a mano che questi si allontana dall'osservatore. Per esempio, se una persona vi saluta con un gesto della mano da distanza ravvicinata, i suoi movimenti vi saranno molto chiari. Viceversa, se la stessa persone vi salutasse a molti metri di distanza, non potreste distinguere i suoi gesti altrettanto chiaramente.

Uno dei metodi più semplici per individuare la posizione di un oggetto in termini di profondità consiste nel far passare l'oggetto stesso davanti o dietro a un'altra figura sullo schermo. Se l'oggetto in questione passa davanti a un altro corpo, come nella Figura 10-4, è ovviamente più vicino allo spettatore. Se invece passa dietro all'altro oggetto è chiaramente più lontano dallo spettatore. Se i due oggetti si scontrano senza che nessuno dei due passi davanti o dietro all'altro, significa che gli oggetti sono equidistanti dallo spettatore.

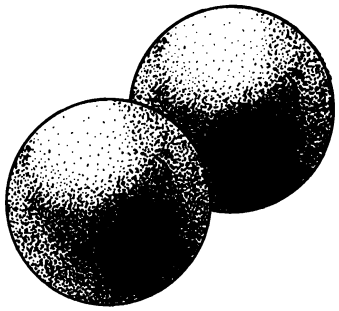


Figura 10-4.

Due figure sovrapposte: la palla a sinistra appare più vicina allo spettatore della palla a destra.

È importante che le tecniche tridimensionali adottate siano compatibili fra loro. Se prima fate retrocedere un oggetto rimpicciolendolo e poi lo fate passare davanti a un altra figura più grande e quindi più vicina allo spettatore, ne risulterà un'immagine confusa, che andrà a scapito dell'efficacia dell'illusione prospettica. Bisogna tenere a mente la posizione di tutti gli oggetti in movimento, in modo da assicurarsi che si incrocino e si sovrappongano l'un l'altro come farebbero degli oggetti reali in un vero spazio tridimensionale.

I vantaggi della prospettiva a tre dimensioni sono evidenti. È molto realistica e riesce a coinvolgere lo spettatore nell'immagine come nessuna altra prospettiva. Lo svantaggio è altrettanto ovvio: la creazione di uno spazio tridimensionale comporta uno sforzo decisamente superiore, anche se è il computer ad addossarsi la parte più difficile.

Prospettiva a due dimensioni e mezzo

La prospettiva a due dimensioni e mezzo rappresenta un compromesso fra la semplice prospettiva 2-D e quella 3-D, più complessa ma anche più realistica. Come nell'animazione a 2-D, anche in quella a due dimensioni e mezzo gli oggetti si muovono solo orizzontalmente su un singolo piano, ma l'esistenza di diversi piani nell'immagine crea l'illusione della profondità.

Per capire come funziona questo tipo di prospettiva, basta pensare a una scena teatrale, con una montagna sottomarina dipinta su un fondale piatto. Questo fondale costituisce uno dei piani dell'immagine presentata al pubblico in platea. Di fronte allo scenario c'è un acquario pieno di pesci, abbastanza largo da occupare l'intero palcoscenico e tanto alto da arrivare in cima al proscenio, ma con uno spessore di soli dieci centimetri. Questo è il secondo piano dell'immagine. Davanti a questo acquario ne viene piazzato un secondo, anch'esso ricco di pesci, che crea il terzo piano dell'immagine. Davanti allo sfondo e ai due acquari viene sistemato un corallo, alto la metà degli acquari. Questo è il quarto piano dell'immagine. La Figura 10-5 (riportata nella pagina successiva) mostra la disposizione dei diversi piani del nostro teatro immaginario.

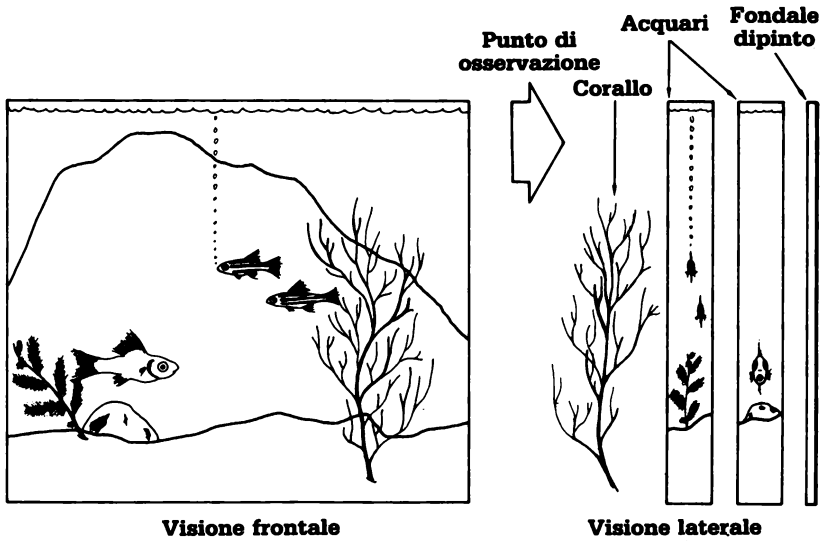


Figura 10-5.

Dimostrazione di prospettiva a due dimensioni e mezzo, creata attraverso l'uso di scenari teatrali e di due acquari particolarmente stretti.

Guardando il palcoscenico dalla platea, si vedrà il corallo davanti a tutto il resto. Dietro il corallo si vedranno pesci che vanno avanti e indietro, mentre alcuni passano davanti ad altri. Dietro ai pesci si vedrà la montagna sottomarina. Pur essendoci solo quattro piani di profondità, sembrerà di vedere un'immagine realmente profonda.

Così come i pesci si possono muovere solo orizzontalmente e verticalmente, e non corrono il rischio di passare da un acquario all'altro, gli oggetti di un'animazione a due dimensioni e mezzo non si allontaneranno né avvicineranno allo spettatore. Ciascun oggetto resterà nel suo piano, passando dietro agli oggetti dei piani antistanti, davanti agli oggetti dei piani retrostanti e scontrandosi con gli oggetti facenti parte del suo stesso piano.

L'animazione a due dimensioni e mezzo è più semplice di quella a 3-D. È più facile calcolare quale oggetto passerà davanti all'altro, e non bisogna preoccuparsi di cambiare le dimensioni, il colore e la quantità di movimento interno degli oggetti.

ANIMAZIONE INTERATTIVA

Nell'animazione tradizionale l'animatore disegna gli elementi di ciascuna immagine, li riunisce per la riproduzione assemblata e registra ogni incremento di movimento in singoli fotogrammi, che proietterà più tardi in rapida successione. Ogni

volta che si proietta il film, si rivede quanto è stato registrato; l'animazione non varia di volta in volta, a meno che il film si rovini o si rompa.

L'animazione computerizzata presenta un grande vantaggio rispetto a quella tradizionale: il computer disegna tutte le immagini e le colloca sullo schermo secondo le nostre istruzioni. Quindi è possibile cambiare rapidamente le modalità dell'animazione, facendo uso di uno strumento di controllo, come il mouse o il joystick, che indichi al computer come modificare il movimento delle immagini, quale parte dello sfondo deve essere visibile e così via. Questo tipo di animazione è detta interattiva (o d'interazione), perché risponde in modo immediato alle richieste dell'operatore e perché l'operatore stesso può interagire immediatamente con le immagini che vede sullo schermo.

L'esempio più comune di animazione interattiva è il puntatore che appare sullo schermo del vostro Amiga. Ogni volta che muovete il mouse l'Amiga sposta il puntatore in una nuova posizione sullo schermo. Potete rispondere alla nuova posizione premendo un tasto del mouse, se il puntatore è giunto all'icona desiderata, oppure muovendo ancora il mouse, per spostare il puntatore nella direzione che interessa.

Altro esempio di animazione interattiva sono i videogiochi, che costituiscono spesso il banco di prova delle capacità di animazione di un computer. Quando si schiaccia il pulsante del joystick per distruggere un'astronave di invasori, il computer risponde con una sequenza animata nella quale l'astronave si trasforma in una palla di fuoco che simula l'esplosione. In genere i giocatori rispondono a loro volta con un aumento del battito cardiaco, un velo di sudore sulla fronte e un crescente coinvolgimento, e maneggiando il joystick con foga tale da rischiare di romperlo.

L'animazione interattiva rappresenta una delle più grandi espressioni artistiche del computer. Coinvolge completamente lo spettatore all'interno dell'azione, come nessun'altra forma di animazione passiva riesce a fare.

L'ANIMAZIONE CON L'AMIGA

L'Amiga è stato appositamente creato per offrire possibilità di animazione di alto livello. I coprocessori possono spostare e manipolare grandi blocchi di dati grafici a una velocità sorprendente, requisito essenziale per una buona animazione computerizzata. Il software sistema, soprattutto grazie a libraries e devices, sfrutta a pieno le capacità grafiche dell'hardware e le rende facilmente accessibili ai programmatori. Lavorando con le diverse applicazioni dell'animazione e sviluppando i vostri programmi BASIC, avrete modo di utilizzare molte delle caratteristiche descritte nelle prossime pagine.

IL CAMPO VISIVO

Probabilmente, pur non conoscendolo con questo nome, saprete già cosa si intende per campo visivo dell'Amiga. Il campo visivo (playfield) è lo scenario grafico di una qualsiasi animazione, esteso anche oltre i limiti fisici dello schermo. Generalmente è immobile e costituisce una sorta di fondale per gli oggetti animati. Quando create immagini fisse con programmi di applicazione grafica quali Graphicraft o Deluxe Paint, state già disegnando figure in un campo visivo. Il cursore o il pennello che muovete sullo schermo è un oggetto mobile, animato dal programma grafico e dai movimenti del mouse.

Come abbiamo visto nel secondo capitolo, l'Amiga può far scorrere il contenuto di una finestra (un campo visivo, appunto) orizzontalmente o verticalmente. I programmatori più esperti potranno avvalersi di questa possibilità per fare una panoramica dello sfondo. Ad esempio, l'animatore può utilizzare come sfondo una visione panoramica di una zona di montagna, molto più estesa della finestra in cui appare, come nella Figura 10-6. Per spostare il punto di osservazione a sinistra basterà far scorrere l'intero campo verso destra, mentre per spostare la posizione dell'osservatore verso destra, sarà sufficiente far scorrere il campo verso sinistra.

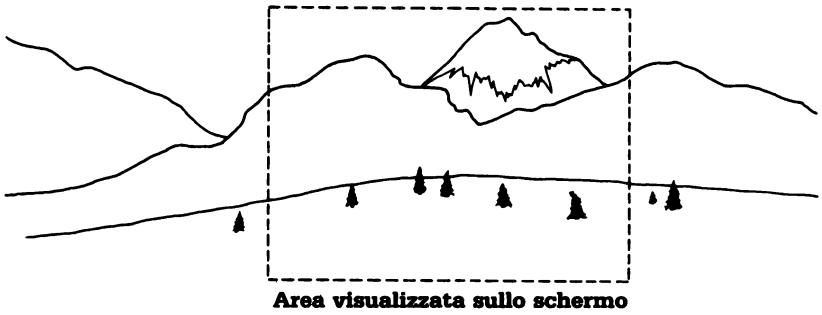


Figura 10-6.

Sfondo panoramico visto attraverso la finestra dell'Amiga. Facendo scorrere l'immagine a destra o a sinistra, si cambia la posizione dell'osservatore immaginario.

Sempre nel secondo capitolo abbiamo visto che l'Amiga può sovrapporre due schermi, grazie al fatto che la trasparenza dello schermo superiore consente di scorgere quello retrostante. In questo modo i programmatori più esperti potranno lavorare con due campi visivi, cosa molto utile soprattutto per l'animazione a due dimensioni e due dimensioni e mezzo. Come nell'esempio del fondale riproducente la montagna sottomarina, il campo visivo posteriore funge da sfondo per gli oggetti che si muovono davanti a esso. Il campo visivo anteriore funziona invece da primo piano, proprio come il corallo precedente. Un programmatore può far passare gli oggetti mobili dietro al campo anteriore, ma essi si muoveranno sempre davanti al campo posteriore. Quando l'operatore fa una panoramica, può far scorrere il campo anteriore più velocemente rispetto a quello posteriore, in modo da far sembrare lo sfondo più lontano e far apparire tutta la scena più realistica.

GEL

Per creare oggetti animati potete utilizzare l'Amiga BASIC o programmi applicativi quali Deluxe Video e Aegis Animator. Tali oggetti, detti gel, possono poi essere spostati all'interno del campo visivo. Gel è un termine proprio dell'animazione Amiga ed è acronimo di Graphic ELement (elemento grafico). Esistono due tipi di gel Amiga: sprite e bob. Gli sprite sono in genere piccoli, semplici e veloci; è sufficiente poco lavoro di elaborazione per farli muovere sullo schermo. I bob possono essere molto più grandi degli sprite, più colorati e più curati nei dettagli, ma occupano un maggiore spazio in memoria; richiedono una notevole mole di lavoro per essere mossi e pertanto risultano talvolta più lenti degli sprite. Le diverse caratteristiche di sprite e bob sono determinate dalle diverse tecniche che l'Amiga impiega per crearli.

Sprite

Uno sprite è un piccolo oggetto collocato nel campo visivo direttamente dal coprocessore Denise. È come un puntatore generato da una penna luminosa utilizzata per indicare i particolari di un'immagine proiettata: quando la penna luminosa si muove verso l'alto, verso il basso o circolarmente, il fascio di luce si muove con essa senza mutare assolutamente l'immagine sottostante. Così come il raggio luminoso della penna, anche lo sprite non è parte integrante dell'immagine disegnata sullo schermo; spostandolo sullo schermo, lo sprite passa sopra il campo visivo senza alterarlo.

Gli sprite sono dispositivi di animazione presenti in diversi

home computer, ad esempio l'Atari 800 e il Commodore 64. Ciò che contraddistingue gli sprite dell'Amiga è il numero e la colorazione. Sull'Amiga è possibile creare fino a otto sprite, ciascuno dei quali può essere di quattro colori differenti. Uno di questi colori è il "trasparente"; attaverso le parti trasparenti degli sprite è possibile intravedere lo sfondo. Per creare sprite con più colori, i programmatori più esperti possono dare all'Amiga istruzioni che combinino due sprite e ne ricavino uno nuovo della stessa misura, che però offra la possibilità di utilizzare sedici colori diversi.

Gli sprite sono facili da creare e richiedono poca memoria per essere registrati. Possono rivelarsi utili come oggetti volanti ad alta velocità nei videogiochi o come puntatori in altre applicazioni. Il puntatore del Workbench è un tipico esempio di sprite.

Bob

A differenza dello sprite, che si sposta sul campo visivo senza alterarlo, il bob è parte integrante del campo stesso. Per capire come il bob si muove sullo schermo, si pensi a un animatore tradizionale che crea una sequenza di fotogrammi disegnando su un unico foglio di carta. Egli disegna un laghetto con un'anatra sulla riva destra e poi ne fa una fotografia. Per far sì che l'anatra nuoti verso il lato sinistro del laghetto, l'animatore cancellerà l'anatra originale per riprodurla pochi millimetri più avanti. Colorerà inoltre il punto del laghetto che era coperto dalla prima anatra. L'animatore dovrà ripetere tutta l'operazione per ogni fotogramma dell'animazione, finché l'anatra non raggiungerà la riva opposta del laghetto.

Un bob inserito in un campo visivo non è altro che l'anatra del nostro esempio. Il blitter (la sezione del coprocessore Agnus riservata al disegno veloce) svolge le funzioni dell'animatore. Disegna, cancella e disegna nuovamente il bob e lo sfondo sottostante, in maniera tanto rapida da far muovere il bob con moto fluido e scorrevole.

I bob possono utilizzare fino a 32 colori alla volta, e possono essere di qualsiasi grandezza, purché contenuti entro i limiti del campo visivo. Dato che, per simulare il movimento, i bob vengono continuamente ridisegnati, il tempo necessario per i calcoli è superiore a quello degli sprite e insorgono difficoltà leggermente maggiori nel corso della programmazione stessa. Nonostante questo i bob si muovono con la stessa rapidità e scorrevolezza degli sprite, a patto che non siano troppo grandi.

I bob sono adatti per la creazione di oggetti animati di forma e colori complessi. Un bob si presta, per esempio, a riprodurre un oggetto grande come un'automobile. Uno sprite non potrebbe creare un oggetto altrettanto grande e curato nei particolari. Un tipico esempio di bob è il pennello custom che selezionate dall'immagine principale del Deluxe Paint. Quando selezionate una sezione dell'immagine da voi creata, la trasformate in pennello e la utilizzate per colorare, non state facendo altro che usare un bob.

CREAZIONE DEL MOVIMENTO INTERNO

Per un programmatore è molto semplice spostare un gel su uno sfondo ricorrendo soltanto al movimento esterno. È sufficiente comunicare all'Amiga la velocità e la direzione desiderate. Esistono due possibilità per caratterizzare un gel anche con un moto interno per renderlo più vivo: un disegno per sequenze o il moto delle componenti.

Disegno per sequenze

Il disegno per sequenze è un metodo di animazione accessibile ai programmatori più esperti che fanno uso delle routine di animazione presenti nel sistema dell'Amiga, oppure a chi usi Deluxe Video in modo avanzato. Alla base del disegno per sequenze c'è la tecnica già presentata all'inizio del capitolo nell'esempio del movimento delle ali di un uccello. Il programmatore disegna una sequenza di singole immagini con differenti stadi di moto interno. Poi riproduce la sequenza in modo da simulare il moto sullo schermo.

Per utilizzare questa tecnica con i gel, l'operatore deve creare una sequenza di forme e colori e utilizzare poi il software dell'Amiga per assegnare la sequenza a un singolo gel. L'Amiga fa ruotare le immagini della sequenza, creando così un gel con tanto di moto interno. Per muovere il gel disegnato mentre è in atto la rotazione delle varie immagini che lo costituiscono, il programmatore non deve fare altro che specificare la distanza, la velocità e la direzione del movimento.

Ad esempio, per simulare il volo di un uccello, il programmatore deve creare una serie di immagini che riproducano un battito d'ali, assegnando poi tutta la serie di disegni a un solo gel. L'Amiga penserà quindi a far ruotare continuamente le immagini a sua disposizione creando comunque un uccello che sbatte le ali. Per far sì che l'uccello si muova rispetto allo sfondo, l'operatore indicherà all'Amiga in che direzione, fino a che punto e a quale velocità muovere il gel, in modo che il moto esterno del volatile si adatti al suo moto interno.

Moto delle componenti

Un esperto programmatore può usare anche una seconda tecnica per la realizzazione del moto interno di un'immagine: il moto delle componenti. Il programmatore crea una serie di piccoli gel e li riunisce poi facendo uso delle routine del software sistema di cui dispone l'Amiga, creando così un'unica figura. Ogni singolo gel, pur avendo il proprio moto esterno (senza alcun moto interno), si muove con le altre parti della figura; in questo modo la serie di gel appare come un'unica figura dotata di moto interno.

Ad esempio, pensate alla sequenza del solito uomo che attraversa la strada. Il corpo è costituito da gel diversi per le gambe, il busto, la testa e le braccia. Ciascun gel ha un proprio moto interno: le braccia e le gambe oscillano avanti e indietro, la testa va su e giù, e così via. La combinazione di tutte queste azioni dà come risultato un uomo che cammina. Questo è il movimento interno della figura. Per aggiungere il movimento esterno e far avanzare l'uomo è sufficiente spostare contemporaneamente i vari gel nella medesima direzione.

Il software sistema dell'Amiga si occupa del complicato processo per il calcolo del moto di ogni singolo gel, quando la figura si muove in una determinata direzione. Il programma deve solo creare i gel che compongono la figura animata, stabilire una serie di movimenti per ognuno di essi e poi indicare la direzione e la velocità desiderati per il moto esterno della figura.

Nel caso si tratti di brevi sequenze animate, la creazione del moto delle diverse componenti può apparire più difficoltosa della creazione di immagini per sequenze. D'altra parte, l'animazione delle singole parti è una tecnica ideale per sequenze lunghe e varie, nonché per l'animazione interattiva in genere, mentre i singoli gel possono variare il loro moto a seconda degli input dell'operatore, diretti o pervenuti attraverso le istruzioni del programma.

PRIORITÀ E COLLISIONI DEI GEL

Cosa succede quando due gel in movimento sullo schermo si scontrano? Si sovrappongono, rimbalzano, oppure vanno in frantumi e prendono fuoco? Le routine sistema riservate all'animazione tengono sotto controllo la posizione di ogni gel all'interno del campo visivo. Se un gel entra in contatto con un altro gel o con i margini del campo, il software sistema emette un segnale che identifica i due gel in collisione o il gel che ha raggiunto il limite del campo visivo. Il programmatore ha la possibilità di fare in modo che i gel rispondano quando si trovano in una determinata situazione: possono allontanarsi, piegarsi, sparire, continuare per la propria strada e realizzare tutte le altre soluzioni immaginabili.

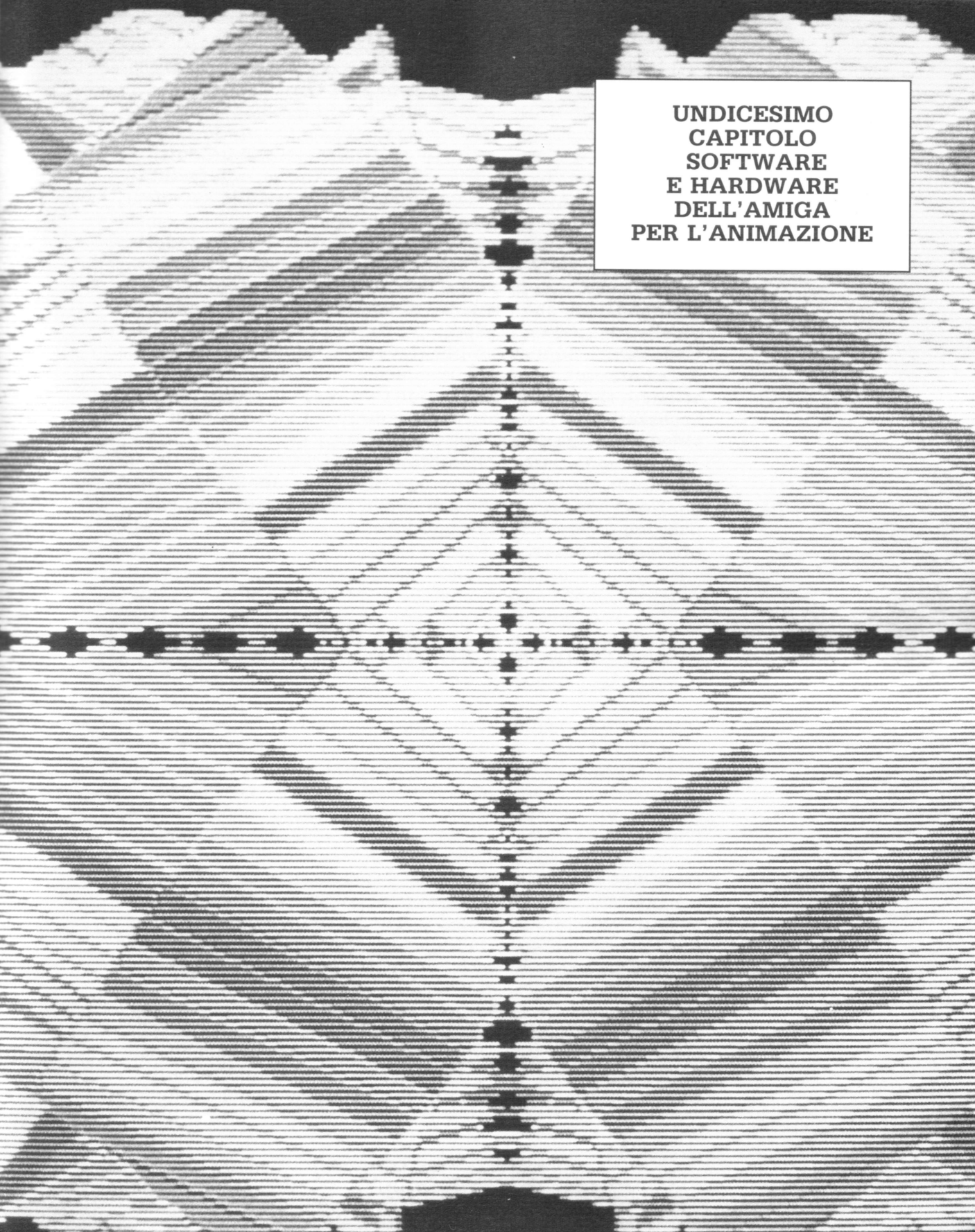
Nell'animazione a tre dimensioni è importante che, quando due gel si incontrano, quello più vicino allo spettatore passi davanti all'altro. Quando un programmatore crea un gel, gli assegna anche un proprio numero di priorità, di cui il software sistema tiene continuamente conto. Un gel con numero di priorità più elevato passerà sempre davanti ai gel con numero inferiore. Per cambiare la posizione di un gel (in termini di profondità), è sufficiente cambiare il numero di priorità, affinché il gel interessato passi davanti e dietro alle figure giuste, creando una perfetta illusione prospettica.

IDEE DI ANIMAZIONE INTERATTIVA PER L'AMIGA

Creare animazioni interattive sull'Amiga è abbastanza semplice, soprattutto grazie a tutte le porte di cui dispone per leggere le informazioni provenienti dal mondo esterno. Le più utili sono le due porte di controllo situate sul lato destro dell'unità centrale. A esse si possono collegare mouse, joystick, tastiere musicali, tavolette grafiche e una vasta gamma di strumenti di controllo che l'Amiga potrà così leggere. Il sistema provvederà a trasmettere le informazioni raccolte da tali strumenti al programma di animazione, in modo che questo possa adattare le sue azioni di conseguenza.

Alcune affascinanti ideazioni di animazione interattiva fanno uso degli elementi hardware più svariati. Ad esempio, aggiungendo un campionatore audio all'Amiga (per informazioni su questi apparecchi, vedere l'ottavo capitolo) che invii al programma di animazione informazioni riguardanti i suoni catturati nello spazio in cui è situato il computer, il programma può adattare le azioni delle figure sullo schermo ai messaggi che riceve, simulando una reazione delle figure stesse ai rumori dell'ambiente circostante. Allo stesso modo, se utilizzate un termometro digitale che, attraverso la porta seriale, invii al computer dati riguardanti la temperatura esterna, il programma che riceve le informazioni potrà creare animazioni differenti a seconda che il tempo sia freddo, fresco, caldo o caldissimo.

Avete avuto modo di constatare che l'animazione sull'Amiga è qualcosa di altamente creativo, che dà possibilità illimitate. Per tenere sotto controllo le miriadi di elementi costitutivi dell'animazione, potete servirvi di un programma applicativo come Deluxe Video o Aegis Animator, oppure creare vostri programmi con l'Amiga BASIC. Nei prossimi tre capitoli vi spiegheremo come fare.



UNDICESIMO CAPITOLO SOFTWARE E HARDWARE DELL'AMIGA PER L'ANIMAZIONE

Il lavoro di animazione non può essere il risultato di un'improvvisazione. Non avviene mai che l'animatore abbia un'idea brillante una sera, lavori qualche ora per realizzare la sua ispirazione e prepari un lavoro da offrire al pubblico il giorno successivo. Grazie però ai prodotti software e hardware disponibili per l'Amiga, che facilitano notevolmente il processo creativo, è possibile avvicinarsi molto a un processo creativo spontaneo.

Il presente capitolo è dedicato proprio a questi prodotti. Nella prima parte descriveremo le tecniche più avanzate che si possono applicare a Deluxe Video, un programma di animazione che consente di creare sequenze animate di eccellente qualità. Sveleremo i trucchi che consentono di visualizzare un numero più elevato di immagini animate, di risparmiare tempo nella creazione di un video e di ottenere risultati ottimi. Infine forniremo le indicazioni necessarie per consentirvi di registrare i video con un videoregistratore una volta ultimati.

La seconda parte del capitolo è dedicata ad Aegis Animator, un altro programma di animazione per l'Amiga. Descriveremo anche i diversi elementi hardware che possono essere utilizzati con l'Amiga per migliorare la sua produzione video.

CONOSCERE DELUXE VIDEO

Deluxe Video è un programma di animazione della Electronic Arts che permette di creare e rivedere sequenze animate di propria ideazione (definite appunto video), con accompagnamento di musica e suoni. Per facilitare l'animazione Deluxe Video lavora su due campi visivi sovrapposti: lo sfondo e il primo piano. Lo sfondo viene utilizzato per creare immagini di fondo fisse e invariabili, mentre le figure animate si muovono sul campo visivo in primo piano. Dal momento che il primo piano è trasparente, tranne nei punti in cui sono visualizzati gli oggetti mobili, i due piani si sovrappongono e si fondono sullo schermo, formando un unico quadro come nell'esempio di animazione a due dimensioni e mezzo che abbiamo visto nel capitolo precedente.

Deluxe Video utilizza tre tipi di animazione. Quella più comune è la cosiddetta "animazione blitter", nella quale i bob visualizzati sullo schermo in primo piano vengono animati dal blitter dell'Amiga, che li disegna, li cancella e li ridisegna continuamente. Il secondo tipo di animazione è quella sequen-

ziale, nella quale Deluxe Video attua la rotazione continua di una serie di immagini create dall'operatore con Deluxe Paint. Il terzo tipo di animazione è il ciclo dei colori, che corrisponde all'animazione realizzata da Deluxe Paint attraverso la rotazione dei registri di colore.

Deluxe Video non è un programma semplice. Deve infatti coordinare il movimento, le variazioni di dimensione, il caricamento del file (da disco), le sparizioni degli oggetti, la musica, gli effetti sonori e molti altri elementi. Per utilizzare efficacemente Deluxe Video bisogna creare le animazioni nello stesso modo in cui un compositore scrive la propria musica, riflettendo attentamente sull'effetto visivo e motorio di ogni particolare e utilizzando tutte le tecniche che si apprendono attraverso l'esperienza pratica della sceneggiatura. I paragrafi successivi spiegano alcune tecniche avanzate e forniscono consigli utili per la realizzazione delle vostre sequenze filmate.

COME UTILIZZARE DELUXE PAINT PER REALIZZARE SFONDI E OGGETTI DI PRIMO PIANO ARMONICI E PROPORZIONATI

Due degli elementi fondamentali che vengono utilizzati in Deluxe Video sono le figure che riempiono lo sfondo e gli oggetti che si muovono in primo piano. Per creare le figure e gli oggetti bisogna usare il programma di applicazione grafica Deluxe Paint, in quanto Deluxe Video non crea oggetti partendo da zero: si limita a caricarli da disco e a manipolarli.

Per creare sul disco un'immagine per lo sfondo da utilizzare con Deluxe Video, è sufficiente salvare su disco un'immagine a bassa risoluzione di Deluxe Paint. Per creare un oggetto da utilizzare nel primo piano, lo si deve ugualmente disegnare su uno schermo di Deluxe Paint a bassa risoluzione, selezionandolo poi come pennello custom e salvandolo su disco. Dal momento che i pennelli custom di Deluxe Paint sono bob (come spiegato nel decimo capitolo) e Deluxe Video utilizza proprio i bob per le sue animazioni, non vi saranno difficoltà nel caricare il pennello da disco e usarlo come figura animata.

Quando si creano oggetti e sfondi per lo stesso video, si cerca di fare in modo che tutte le componenti si accordino per dimensione, stile e accostamenti cromatici. Realizzando ogni oggetto e ogni figura in un momento diverso, si rischia di rimanere delusi dal risultato finale quando tutte le parti appaiono contemporaneamente sullo schermo. È possibile utilizzare Deluxe Paint in modo da assicurarsi che lo sfondo e gli oggetti riportati su Deluxe Video costituiscano un insieme ben amalgamato, per evitare di perdere tempo in rifacimenti.

Gli accostamenti cromatici

Il segreto per ottenere un insieme di colori di un certo gusto consiste nel tenere ben presente quanti colori possono essere impiegati nei due piani di Deluxe Video. Definiamo immagini le figure che compàiono sullo sfondo. Esse possono utilizzare un massimo di otto colori, dal momento che il campo visivo di Deluxe Video ha una profondità di tre bit-plane. Definiamo invece oggetti le figure di primo piano. Anche lo schermo del primo piano è formato da tre bit-plane (diversi da quelli dello sfondo); quindi, qualcuno penserà immediatamente, anche per gli oggetti animati si possono usare otto colori. Invece il numero è limitato a sette perché l'ottavo colore (lo sfondo del campo visivo in primo piano) è necessariamente trasparente: attraverso il primo piano si deve vedere lo sfondo. Dal momento che sfondo e primo piano possono utilizzare serie di colori diversi, sovrapponendoli otterrete una gamma cromatica complessiva di 15 tinte.

Quando si creano oggetti e immagini per un video, è possibile selezionare le tonalità per Deluxe Paint in modo da utilizzare un gruppo di otto tinte per lo sfondo e un altro di sette per il primo piano. Dato che Deluxe Paint ha uno schermo di riserva, è possibile lavorare alle immagini per lo sfondo su uno schermo e agli oggetti di primo piano sull'altro. Le istruzioni seguenti spiegano come creare un'immagine e degli oggetti da utilizzare poi per Deluxe Video.

Innanzitutto attivate Deluxe Paint:

1. Caricate Deluxe Paint immettendo il comando **dpaint lo 4** (con Deluxe Paint II occorre utilizzare il requester introduttivo). Verrà caricata una versione di Deluxe Paint che utilizza quattro bit-plane e offre la possibilità di lavorare con 16 colori. Quando appare lo schermo di Deluxe Paint, vengono visualizzate in fondo alla striscia di controllo due colonne di otto colori. Utilizzate le tonalità sulla sinistra per le immagini dello sfondo e quelle sulla destra per gli oggetti di primo piano.

2. Aprite la palette per cambiare i colori disponibili a vostro piacimento.

3. Inserite nella colonna di sinistra le otto tinte che volete adoperare per lo sfondo.

4. Inserite nella colonna di destra i sette colori che volete utilizzare per gli oggetti. Dal momento che l'ottavo colore,

destinato allo sfondo, viene considerato trasparente da Deluxe Video, all'ultimo posto della colonna di destra potete inserire un colore qualsiasi. È naturalmente preferibile che questa seconda serie di colori, pur essendo diversa, si accompagni bene alla prima gamma cromatica selezionata nella colonna di sinistra. Questo accorgimento darà una colorazione più vivida all'immagine finale, quando sfondo e oggetti verranno assemblati.

5. Chiudete la palette premendo il tasto **OK**.

Ora che Deluxe Paint è pronto, si può passare alla creazione dello sfondo:

1. Disegnate l'immagine utilizzando i colori della colonna di sinistra.

2. Selezionate **Save As...** dal menu **Picture** per salvare l'immagine dello sfondo su disco. Per salvare la figura direttamente sul disco Deluxe Video Parts, eseguite i punti seguenti:

3. Inserite il disco Deluxe Video Parts in un disk drive.

4. Quando appare la scritta **Save**, selezionate col mouse il riquadro **Drawer** e utilizzate il tasto DELETE per cancellarne il contenuto. Se il disco Parts è in un drive esterno, inserite prima il nome del drive (ad esempio **df1:**) e poi **Pictures**. Per immettere il nome dell'immagine dello sfondo, selezionate il riquadro **File** e scrivete il nome. Premete il tasto **Save** per salvare l'immagine nel cassetto **Pictures** del disco Deluxe Video Parts.

Dopo aver creato l'immagine dello sfondo, si possono disegnare gli oggetti che volete abbinarle:

1. Premete il tasto **j** per passare al secondo schermo di Deluxe Paint.

2. Utilizzate il bottone di destra del mouse per selezionare l'ultimo colore della colonna di destra come tinta dello sfondo, poi premete il tasto **CLR** per coprire l'intero schermo con questo colore. Dal momento che lo sfondo, su Deluxe Video, risulterà trasparente, non importa quale colore abbiate scelto.

3. Disegnate gli oggetti sullo schermo utilizzando i primi sette colori della colonna di destra.

4. Per controllare le dimensioni e il colore degli oggetti rispetto allo sfondo, utilizzate l'apposito comando per selezionare l'oggetto come pennello. Adoperate il mouse per muovere l'oggetto sullo sfondo e vederne l'effetto in movimento. Al termine, tornate allo schermo degli oggetti.

5. Quando siete pienamente soddisfatti dell'oggetto creato, selezionatelo nuovamente come pennello.

6. Immettete il comando **Save As...** dal menu **Brush** per memorizzare l'oggetto.

7. Ripetete la procedura di registrazione vista per l'immagine dello sfondo, ricordandovi solo di salvare il pennello nel cassetto **Objects** del disco Deluxe Video Parts.

Se dovete creare nuove immagini e nuovi oggetti, è possibile scegliere nuovi colori per la palette e utilizzare le tecniche già viste per creare nuove immagini di sfondo con i relativi oggetti. Dopo aver creato e salvato immagini e oggetti, uscite da Deluxe Paint e passate al programma Deluxe Video Maker. Per inserire in Deluxe Video le immagini e gli oggetti appena creati, procedete secondo le istruzioni riportate qui di seguito:

1. Aprite lo scenario in cui volete inserire l'immagine di sfondo.

2. Aggiungete una pista **Picture** per l'immagine in questione. Quando appare la richiesta di specificare quale figura si vuole caricare, digitate il nome con cui avete precedentemente salvato l'immagine.

3. Apparirà la scritta **Palette More Than 8 Colors** (palette con più di 8 colori). Infatti, Deluxe Video sa che c'erano 16 colori nella palette utilizzata da Deluxe Paint, anche se poi i colori usati per lo sfondo sono solo otto. Selezionate l'opzione **Best**; Deluxe Video caricherà l'immagine dello sfondo utilizzando i colori originali.

4. Puntate sulla scritta **Palette More Than 8 Colors** e premete il tasto **Select** per completare l'aggiunta della pista **Picture**.

5. Per caricare gli oggetti relativi alla figura di sfondo già predisposta, aggiungete una pista **Object** per ogni oggetto. Quando appare la richiesta per ogni pista, selezionate l'oggetto creato, poi l'opzione **Best** per caricarlo mantenendo i colori originali, e infine premete **Select**.

Utilizzando le tecniche presentate in questo paragrafo è possibile mantenere immagini e oggetti in sintonia ed evitare così di creare una nuova palette per ottenere buoni accostamenti fra oggetti e immagini create con palette completamente diverse.

IL CICLO DI COLORI PER LE IMMAGINI DELLO SFONDO

Le animazioni di Deluxe Video avvengono per la maggior parte nel primo piano, con oggetti mossi dai comandi **Move To** di "sceneggiatura". Questo tipo di movimento è una semplice animazione blitter che sposta i bob sullo schermo in primo piano. L'animazione blitter richiede grandi quantità di tempo e di memoria ai processori, e più sono gli oggetti che si aggiungono sullo schermo, più i loro movimenti si fanno impacciati. Se si vuole impiegare un metodo di animazione diverso, che richiede poca memoria e poco tempo di realizzazione e permette anche di aggiungere animazione allo sfondo, si può utilizzare l'animazione basata sul ciclo di colori.

Probabilmente avete già lavorato con il ciclo dei colori in Deluxe Paint. Scegliendo una gamma cromatica come gamma di rotazione, visualizzando sullo schermo una successione dei colori che la compongono e facendo infine ruotare le tinte, si riesce a dare l'illusione del movimento. È possibile utilizzare il ciclo dei colori anche in Deluxe Video, selezionando i colori che ruoteranno sia sul primo piano che sullo sfondo. Si può usare Deluxe Paint allo scopo di creare immagini o oggetti da usare poi con l'effetto color-cycling di Deluxe Video.

L'esempio seguente utilizza il ciclo dei colori. All'inizio viene creata un'immagine di sfondo che rappresenta un tiro a segno da luna park, con una serie di papere bersaglio che si muovono su un nastro trasportatore dietro la tenda. Per creare questa figura come sfondo, è necessario caricare prima Deluxe Paint e definire la palette.

1. Immettere **dpaint lo 4** per caricare Deluxe Paint con quattro bit-plane in modo da avere 16 colori nella palette (con Deluxe Paint II occorre utilizzare il requester introduttivo).

2. Premere il tasto **p** per aprire la palette e poter selezionare i colori.

3. Le otto tonalità nella colonna di sinistra vengono usate per l'immagine dello sfondo. Lasciare come primi tre colori il nero, il bianco e il rosso, ma sostituire il quarto con il marrone. Queste quattro tinte verranno usate per creare la tenda e un bancone di legno.

4. Gli ultimi quattro colori dello sfondo verranno utilizzati per creare le papere. Di essi, il primo deve essere il giallo. Lasciate per ora gli altri tre come sono (verranno cambiati in seguito).

5. Definite gli ultimi quattro colori dello sfondo (il giallo appena selezionato e i tre sottostanti) come ciclo di colori numero uno (**C1**) (vedere il terzo capitolo per maggiori informazioni se non si conosce bene la procedura).

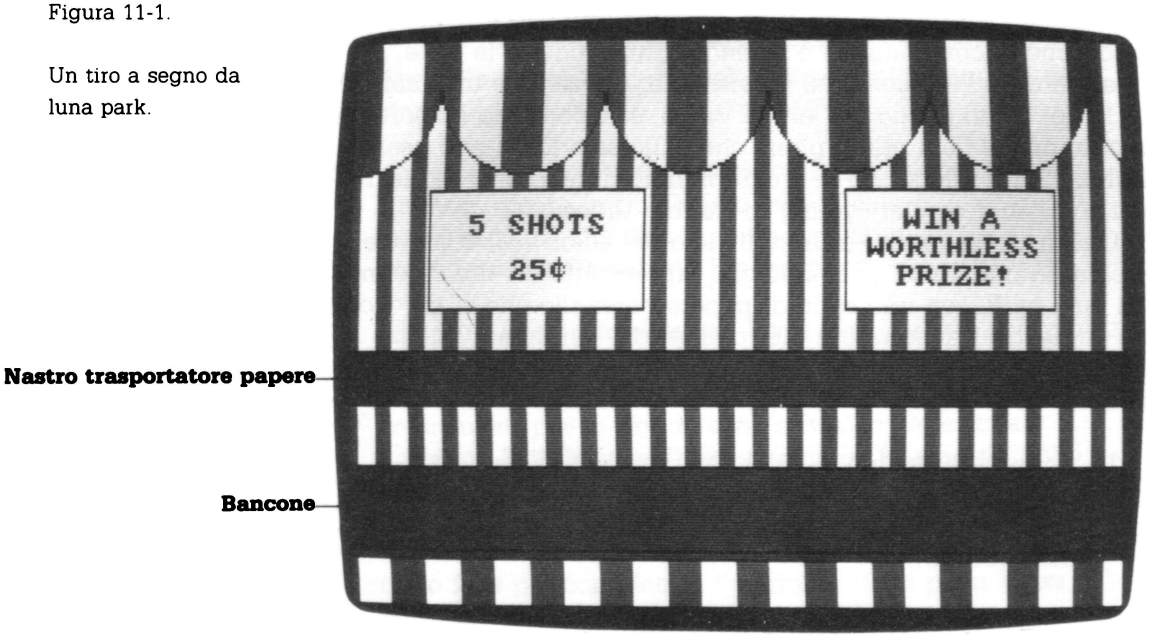
6. Chiudere la palette selezionando **OK**.

Ora si possono creare la tenda e le papere:

1. Utilizzate i primi quattro colori della colonna di sinistra per disegnare una tenda con il bancone e il nastro trasportatore per le papere bersaglio, come illustra la Figura 11-1. La striscia marrone nel mezzo dell'immagine è il nastro trasportatore delle papere.

Figura 11-1.

Un tiro a segno da luna park.



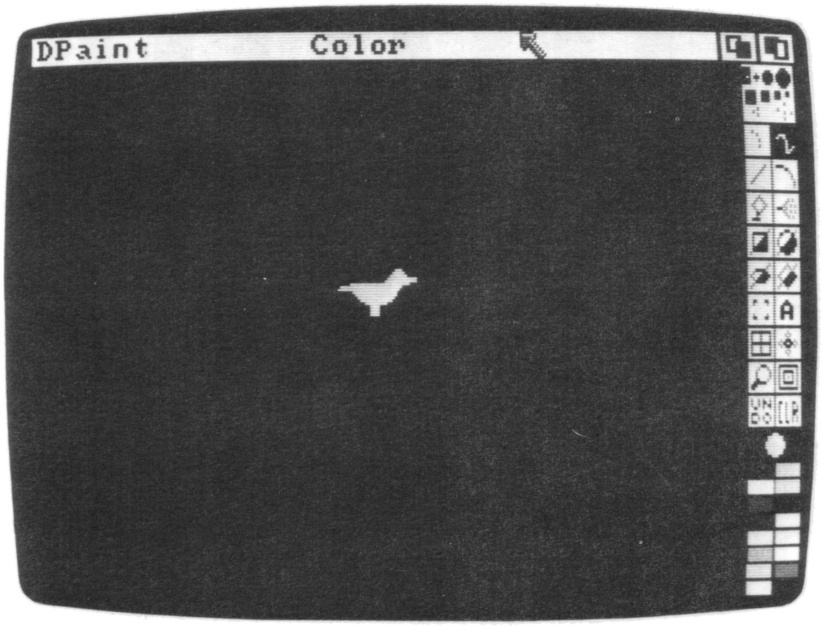


Figura 11-2.

La papera selezionata come pennello.

2. Passate al secondo schermo e disegnate una papera di colore giallo, come mostra la Figura 11-2.

3. Utilizzate l'apposito comando per selezionare la papera come pennello e tornate quindi all'immagine del tiro a segno.

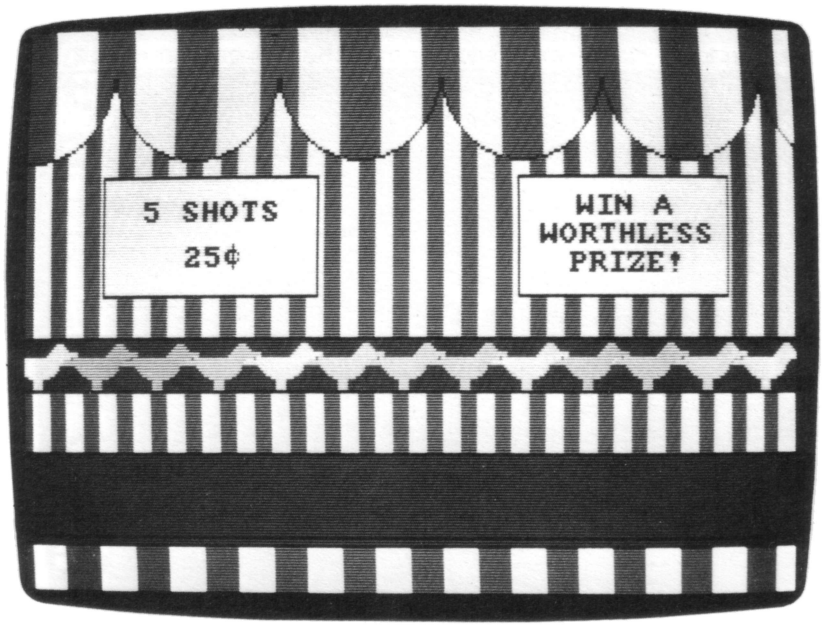
4. Selezionate **Cycle** dal menu **Mode**, in modo che il pennello faccia ruotare le tinte della gamma cromatica fissata precedentemente nella palette.

5. Selezionate il primo dei quattro colori cycle (il giallo) come colore di primo piano. In questo modo il pennello inizierà la rotazione a partire da quel colore.

6. Disegnate una fila di papere attaccate una all'altra, col becco rivolto verso destra, lungo tutto il nastro trasportatore; premete il bottone del mouse una volta per ogni papera disegnata. Ogni volta che premete, la papera dovrebbe cambiare colore, perché il pennello esegue il ciclo dei colori. Alla fine si dovrebbe ottenere un risultato simile alla Figura 11-3 (riportata nella pagina successiva).

Figura 11-3.

Una fila di papere disegnate utilizzando i colori della gamma cycle e posizionate nell'apposita striscia.



Se a questo punto fate ruotare i colori premendo il tasto TAB, le papere resteranno immobili e cambieranno continuamente colore. Per dare la sensazione del movimento, bisogna modificare nuovamente la palette:

1. Aprite la palette.

2. Copiate negli ultimi tre colori della gamma scelta per la rotazione (sesto, settimo e ottavo colore) il marrone che avete usato per il nastro trasportatore del tiro a segno.

3. Selezionate **OK** per chiudere la palette.

4. Premete il tasto TAB per controllare. Dovrebbe apparire una serie di papere gialle che si muovono lungo il nastro trasportatore a distanza regolare.

5. Salvate l'immagine nel cassetto **Pictures** del disco Deluxe Video Parts.

A questo punto, per creare qualche oggetto in primo piano che abbellisca l'immagine della tenda sullo sfondo (popcorn, premi, fucili giocattolo), bisogna passare al secondo schermo, per poter disegnare gli oggetti e salvarli nel cassetto **Objects** del disco Deluxe Video Parts. Al termine, uscite da Deluxe Paint, passate a Deluxe Video e inserite la vostra immagine col suo ciclo di colori.

1. Utilizzate le tecniche descritte precedentemente per inserire la figura in uno scenario usando una pista **Picture**.

2. Fate riapparire la figura immettendo un effetto **Load** nella pista **Picture**.

3. Aggiungete alla scena una pista **Background**.

4. Aggiungete un effetto **CycleClr** alla pista **Background**.

5. Quando appare la richiesta **Cycle Colors**, assicuratevi che i colori 4, 5, 6 e 7 siano attivati a che i colori 0, 1, 2 e 3 siano spenti; i colori attivati sono riconoscibili per il numero illuminato.

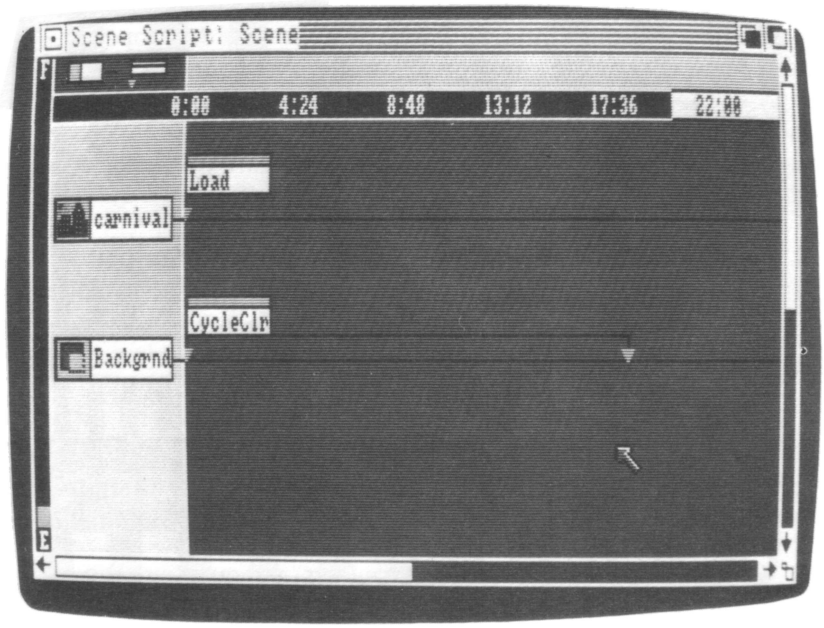
6. Premete il tasto **OK** per accettare la richiesta appena effettuata.

7. Spostate la prima freccia dell'effetto **CycleClr** sul tempo in cui volete che inizi il movimento delle papere, e spostate la seconda freccia sul tempo prescelto per la fine dell'azione. La Figura 11-4 mostra una sceneggiatura con l'immagine caricata e il ciclo di colori fissato su una durata di 18 secondi.

8. Fate girare la scena in modo da verificare la velocità delle papere. Se sono troppo veloci, puntate l'effetto **CycleClr** con il mouse e premete due volte il bottone per riaprirlo e fissare il regolatore **Speed** alla velocità voluta.

Figura 11-4.

Un copione predisposto per l'animazione a ciclo di colori.



Una volta creato lo sfondo con le papere mobili, potete continuare ad aggiungere oggetti animati su una pista **Foreground** davanti allo sfondo ed effetti sonori su una pista **Sound**, allo scopo di aumentare la spettacolarità del video.

La tecnica dell'animazione a ciclo di colori può essere usata nei modi più svariati per ottenere sfondi di eccellente qualità visiva. Questa tecnica può essere sfruttata per simulare fiumi rigogliosi, alberi e prati spazzati dal vento, folle ondeggianti o qualsiasi altro movimento ripetitivo di sottofondo. Quale che sia l'azione svolta dagli oggetti in primo piano, lo sfondo retrostante continuerà in ogni caso a svolgere la sua funzione senza il minimo intoppo.

CREAZIONE DI ANIMAZIONE A SEQUENZE DI DISEGNI

L'animazione sequenziale è l'animazione più sofisticata che potete realizzare con Deluxe Video, ma è anche quella che comporta il maggior impiego di tempo. Per creare la serie di immagini sequenziali richieste da questo tipo di animazione, bisogna lavorare con tre programmi diversi: Deluxe Paint per disegnare le figure, Deluxe Video Framer per fondere i vari fotogrammi in un unico oggetto e Deluxe Video Maker per inserire l'oggetto in un video e farlo muovere attraverso la rapida successione delle sequenze. Le istruzioni seguenti spiegano come realizzare un oggetto animato sequenzialmente con un lavoro non eccessivamente complicato e prolisso.

Disegnate innanzitutto l'intera sequenza con Deluxe Paint:

1. Caricate Deluxe Paint immettendo il comando **dpaint lo 3** (con Deluxe Paint II occorre utilizzare il requester introduttivo), in modo da ottenere una versione con tre bit-plane che metta a vostra disposizione otto colori.

2. Selezionate negli ultimi sette colori della palette le tinte che volete usare per la vostra immagine. Il primo colore della palette fungerà da colore di sfondo; non importa quindi quale tonalità scegliate, perché Deluxe Video, riconoscendola come sfondo, la farà risultare comunque trasparente.

3. Disegnate la prima figura nell'angolo superiore sinistro dello schermo, in modo da lasciare spazio sufficiente per le altre immagini che andranno a comporre la sequenza. Assicuratevi che l'immagine sia abbastanza distanziata dalla linea d'intestazione. Se la disegnate troppo vicina, non riuscirete a incorniciarla con Deluxe Video Framer: sarà troppo alta e impedirà l'inserimento della cornice.

4. Tracciate un riquadro che racchiuda l'immagine, come mostra la Figura 11-5.

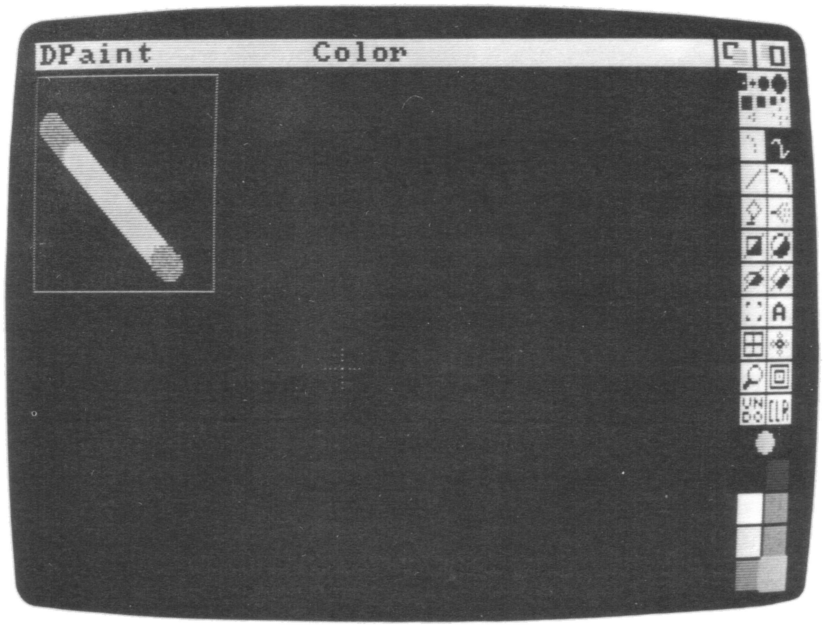


Figura 11-5.

Un'immagine contornata pronta per essere duplicata in modo da creare una sequenza.

5. Utilizzate l'apposita funzione per selezionare l'intero riquadro come pennello.

6. Riproducete più volte l'immagine originale, come mostra la Figura 11-6. Fate attenzione a non sovrapporre i margini dei diversi riquadri: lo spessore delle linee che separano le figure deve essere doppio rispetto allo spessore del perimetro esterno. Quando avete riempito lo schermo in tutta la sua lunghezza, tornate sotto la prima figura a sinistra e ricominciate.

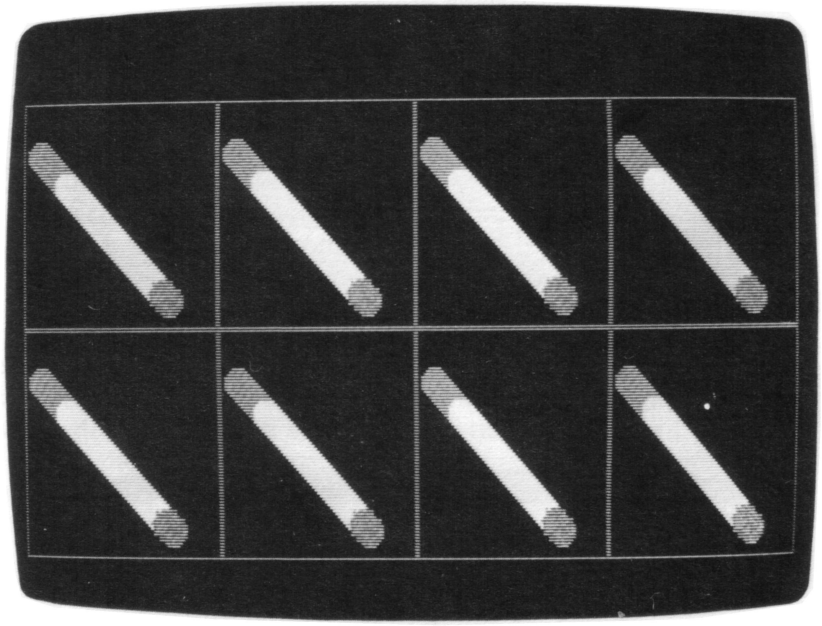


Figura 11-6.

Una sequenza di immagini pronte per essere modificate.

7. Modificate le immagini in modo da ottenere una sequenza di immagini che si sviluppi da sinistra a destra e dall'alto verso il basso. Il riquadro che racchiude l'immagine non è altro che il campo in cui si svolgerà l'azione delle immagini in sequenza; utilizzatelo come punto di riferimento per evitare di modificare troppo la figura fra un fotogramma e l'altro. La Figura 11-7 mostra un esempio di sequenza di immagini.

8. Quando avete finito di disegnare, selezionate il comando **Save As...** dal menu **Picture** per salvare l'intera sequenza nel cassetto **Pictures** del disco Parts.

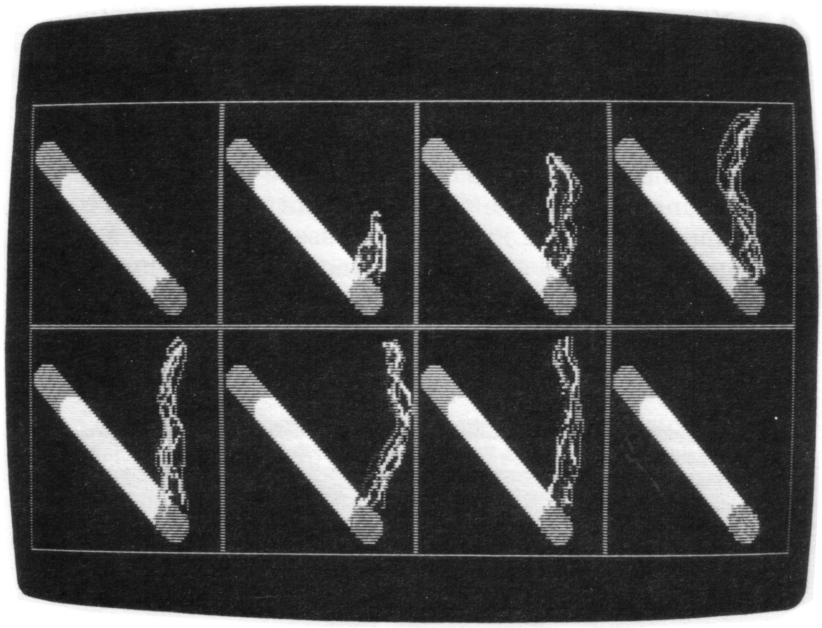


Figura 11-7.

Una sequenza di immagini progressive.

Dopo aver registrato la sequenza sotto forma di immagine, potete uscire da Deluxe Paint e caricare Deluxe Video Framer, per trasformare finalmente l'immagine in una sequenza animata:

1. Fate girare il Framer.

2. Caricate l'immagine appena realizzata.

3. Selezionate **Change Frame...** dal menu **Project**.

4. Quando appare la richiesta **Frame Sequence**, immettete nella parte sinistra il numero dei riquadri contenuti sullo schermo in senso orizzontale, e a destra il numero dei riquadri in senso verticale. Nel nostro esempio le cifre sarebbero rispettivamente 4 e 2. Al termine selezionate **OK** per confermare.

5. A questo punto apparirà una griglia. Fate coincidere l'angolo superiore sinistro di questa griglia con lo stesso angolo dei riquadri contenenti le immagini.

6. Spostate l'angolo inferiore destro del riquadro contrassegnato dalla lettera **"a"** verso il basso e verso destra, in modo che la griglia vada a coincidere esattamente con i margini dei riquadri.

7. Le immagini verranno contrassegnate da lettere poste nell'angolo superiore sinistro di ogni sezione della griglia, come risulta dalla Figura 11-8. Osservate ciò che succede in ogni riquadro. Più tardi, quando userete Deluxe Video Maker, potrete cambiare l'ordine delle immagini in sequenza immettendo semplicemente le lettere nell'ordine desiderato.

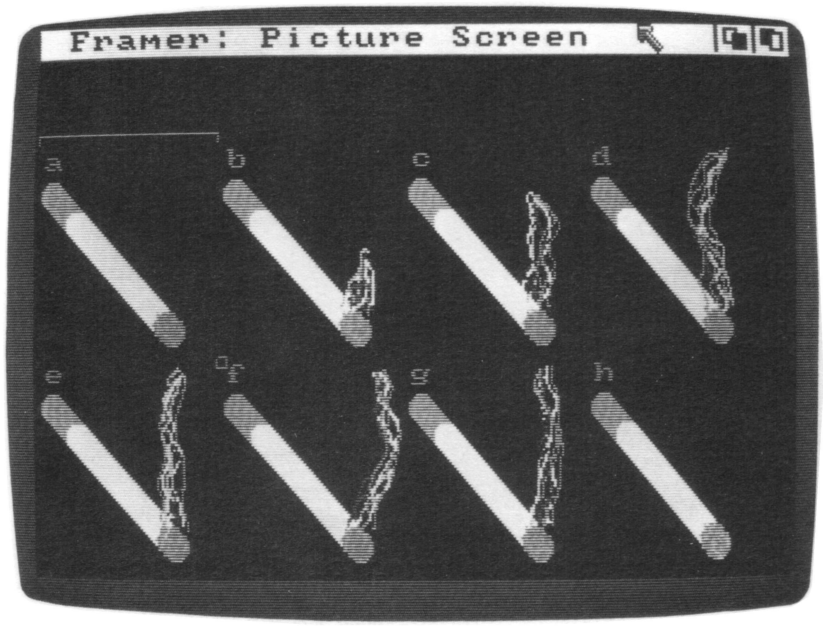


Figura 11-8.

Una sequenza di immagini incorniciate e numerate da Deluxe Video Framer.

8. Selezionate **Make Object From Frame** dal menu **Project** per fondere le immagini in una sequenza animata.

9. Posizionate l'unico riquadro rimasto sullo schermo nel punto in cui volete che si svolga l'azione. Selezionate quindi **Animate** dal menu **Project** e il Framer eseguirà la sequenza da voi creata.

10. Selezionate il comando **Save...** dal menu **Object** per salvare la sequenza, sotto forma di oggetto, nel cassetto **Objects** del disco Parts. Notate come il Framer aggiunga al nome del vecchio file l'estensione **-1**: selezionate **Save** e la sequenza sarà memorizzata sotto quel nome.

Una volta realizzata la sequenza con il Framer, passate a Deluxe Video Maker per inserirla in una sorta di copione cinematografico:

1. Fate girare Deluxe Video Maker.

2. Aggiungete una pista **Object** alla scena.

3. Quando appare la richiesta **Object**, selezionate il nome della sequenza che avete appena realizzato.

4. Aggiungete un effetto **Appear** alla pista **Object**.

5. Quando appare la richiesta **Appear Where?**, posizionate il riquadro nel punto in cui volete che appaia e selezionate **OK** per confermare.

6. Aggiungete un effetto **AnimSeqn** alla pista **Object** subito dopo l'effetto **Appear**.

7. Quando appare la richiesta **Animate Sequence**, vengono visualizzati i nomi (di una lettera) di ogni immagine contenuta nella sequenza. Andate a rivedervi (se ve lo siete annotato) l'ordine in cui volete fare apparire i fotogrammi, dopodiché immettete le lettere nella sequenza desiderata nel riquadro **Sequence**, cancellando prima il suo contenuto. Se volete che il programma ripeta più volte la sequenza appena immessa, fissate il numero **Repetitions** a vostro piacimento e selezionate **OK** per confermare.

8. Spostate la freccia sinistra dell'evento **AnimSeqn** sul tempo del copione in cui desiderate che inizi l'azione, e la freccia destra nel punto in cui volete che l'animazione termini. Deluxe Video farà ruotare le immagini impiegando tutto il tempo fissato dall'operatore. Per cui, se fissate un tempo di esecuzione molto lungo, la sequenza farà scorrere le immagini molto lentamente, mentre un breve tempo di esecuzione significherà una veloce rotazione dei fotogrammi.

9. Fate girare la scena per osservare la sequenza. Se il tempo che avete fissato nell'effetto **AnimSeqn** fa ruotare le immagini troppo lentamente o velocemente, potete riaprire l'effetto stesso e aumentare o diminuire il numero di immagini o il numero di ripetizioni.

L'animazione sequenziale può essere utilizzata per ottenere effetti strabilianti. Ricordatevi che mentre l'oggetto sta facendo ruotare i fotogrammi che lo compongono, potete farlo muovere con un effetto **Move To**. Se create un oggetto che vola o che cammina, è possibile coordinarne il movimento esterno (definito con l'effetto **Move To**) con il movimento interno (definito con l'effetto **AnimSeqn**), dando l'impressione che l'oggetto stia veramente volando o camminando.

SINCRONIZZAZIONE DEGLI EFFETTI

Nell'usare Deluxe Video scoprirete che è molto importante sincronizzare i vari effetti, in modo che il suono accompagni le giuste azioni, il movimento inizi nello stesso istante in cui l'oggetto appare, e altri effetti di qualsiasi genere siano perfettamente sincronizzati. Un modo piuttosto semplice di ottenere tale sincronismo consiste nell'avviare tutti gli effetti allineando le frecce sinistre dei rispettivi riquadri. Mentre spostate una freccia lungo la sua pista, è possibile leggere il tempo relativo nella parte superiore del copione.

Può capitare che si verifichi l'esigenza di fissare l'inizio di un determinato effetto in modo che avvenga in un punto preciso nel corso dell'azione svolta da un altro effetto. Ad esempio, si potrebbe aggiungere il suono di una bottiglia stappata nel bel mezzo di una sequenza che mostra appunto qualcuno che stura una bottiglia. Come si fa a trovare il punto esatto all'interno della sequenza, risparmiandosi la noia di tentare a casaccio e verificare ogni volta? Per conoscere il tempo esatto in cui viene aperta la bottiglia, potete ricorrere al controllo remoto.

Quando fate girare un video o una scena col controllo remoto visualizzato sullo schermo, nella parte inferiore della schermata appare il tempo di esecuzione del copione, tempo che va avanti e indietro a seconda delle parti di copione che visualizzate. Per trovare l'esatta collocazione di un certo effetto all'interno della scena, potete bloccare l'azione e leggere il tempo riportato dal controllo remoto. Per facilitare la ricerca è consigliabile iniziare la riproduzione da un punto vicino alla scena che vi interessa, procedendo poi fotogramma per fotogramma.

Per procedere in questo modo (detto modo single-step, cioè "un passo alla volta"), selezionate il pulsante del controllo remoto contenente tre punti, dopodiché utilizzate il pulsante della riproduzione in avanti o all'indietro: vi muoverete di un fotogramma alla volta. Quando avete trovato il punto della scena che vi interessa, annotate il tempo visualizzato dal controllo remoto. Tornate quindi al copione e fissate il nuovo effetto sul tempo che avete annotato, posizionando la freccia sinistra nella giusta locazione, indicata dal tempo che scorre nella parte superiore del copione.

Quando utilizzate questo sistema, assicuratevi che nel menu

Options non sia stata selezionata l'opzione **Realtime**. In caso contrario, il tempo visualizzato dal controllo remoto continuerà ad aumentare tanto durante l'operazione di caricamento dei programmi quanto durante gli effetti video, con lo spiacevole risultato che il tempo riportato non avrà nulla a che vedere con la locazione temporale degli effetti all'interno del copione.

COME SCRIVERE COPIONI LEGGIBILI

Quando realizzate i copioni per i vostri video, lo schermo viene riempito da una moltitudine di piste ed effetti, che aumentano in relazione al crescere degli effetti riprodotti nel contempo su una pista. Se volete revisionare il copione, può risultare alquanto difficile districarsi nella confusione: alcuni riquadri potrebbero averne completamente coperti altri, le frecce potrebbero essere ammucchiate in spazi ridotti, impedendo di distinguere chiaramente l'inizio e la fine dei vari effetti; infine, alcuni riquadri potrebbero essere lontani dalle rispettive frecce tanto da rendere quasi impossibile il riconoscimento degli abbinamenti.

Per evitare questi problemi, ecco qui di seguito alcuni consigli utili per rendere i vostri copioni facili da leggere e revisionare:

1. Mantenete le piste collegate vicine fra loro. Sarà più facile sincronizzare gli effetti fra le piste e seguire le piste collegate all'interno del copione. Ad esempio, se volete riprodurre due uccelli che volano e gracchiano, dovete utilizzare tre piste: due piste **Object** (una per ogni uccello) e una pista **Sound**. Fate in modo di tenere queste tre piste unite nel copione, per facilitare successive sincronizzazioni.

2. Se avete molti effetti a esecuzione simultanea riuniti in una sola pista, disponeteli uno sopra l'altro in modo che ciascun tipo di effetto vada a occupare un livello ben preciso al di sopra o al di sotto della linea centrale della pista. Questa disposizione vi consente di seguire lo svolgimento di ogni singolo effetto e di vedere l'interazione che collega un effetto all'altro. Ad esempio, in una pista **Object** potreste avere diversi effetti **AnimSeqn**, eseguiti contemporaneamente a effetti **Move To** e a effetti **Size**. Per conservare una certa chiarezza può essere utile mettere tutti i riquadri degli effetti **AnimSeqn** molto vicini alla linea della pista. Al livello immediatamente superiore agli effetti **AnimSeqn** potete sistemare gli effetti **Move To**, e ancora sopra possono essere collocati i riquadri degli effetti **Size**, come mostra la Figura 11-9.

3. Se l'eccessivo raggruppamento di effetti impedisce una chiara lettura dei singoli riquadri, aumentate le dimensioni del copione utilizzando il comando **Time Scale...** nel menu **Options**. In questo modo avrete più spazio per sistemare ordinatamente tutti gli effetti senza sovrapporre i riquadri o nascondere le frecce. Ad esempio, se il copione è predisposto per visualizzare gli effetti di 22 secondi di animazione, e i riquadri si addensano eccessivamente intorno alle piste, potete predisporre il copione in modo che visualizzi solo 7 secondi di animazione, triplicando in pratica lo spazio disponibile su ogni pista per sistemare tutti gli effetti.

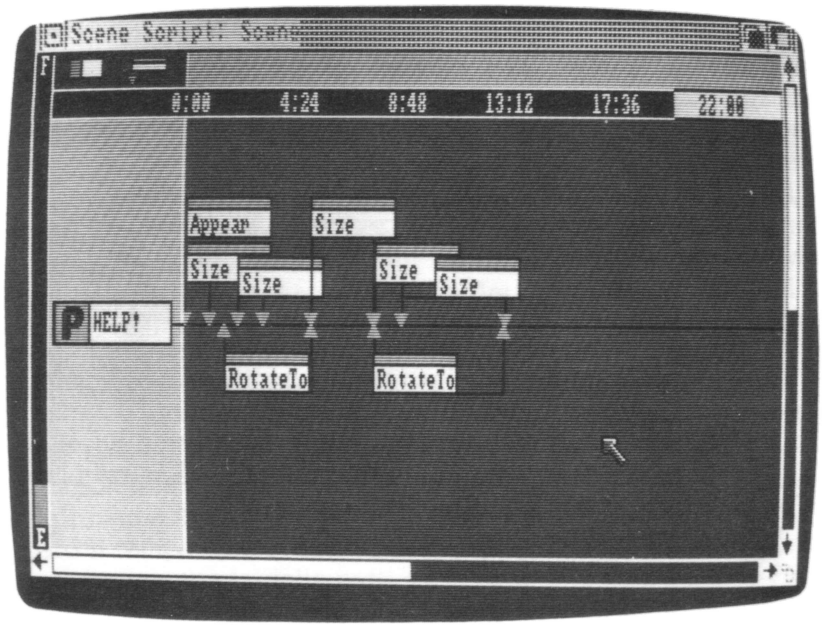


Figura 11-9.

Una traccia Object lungo la quale gli effetti sono raggruppati su livelli diversi a seconda del tipo, in modo da rendere il copione più leggibile.

VIDEOREGISTRAZIONE DELLE ANIMAZIONI DI DELUXE VIDEO

Quando avete ultimato la creazione di un video con Deluxe Video, potete registrarlo su videocassetta. È molto semplice: basta collegare l'Amiga a un videoregistratore, avviare la registrazione e far girare la sequenza animata. I paragrafi seguenti forniranno tutte le informazioni necessarie.

Collegamento dell'Amiga a un videoregistratore

Per connettere l'output video dell'Amiga all'input video di un videoregistratore è sufficiente un cavo audio con spine phono su entrambe le estremità del filo. Inserite una spina nella porta video posta sul retro dell'unità centrale dell'Amiga, e l'altra spina nella presa contrassegnata dalla scritta "Video In" (o qualcosa del genere) posta sul retro del videoregistratore.

Se la sequenza animata ha anche un accompagnamento sonoro, bisogna collegare anche le porte audio dell'Amiga alle prese "Audio In" del videoregistratore. Se quest'ultimo è stereo, collegate l'Amiga alle prese del canale destro e sinistro nello stesso modo in cui collegheresteste un normale registratore a cassette (vedere l'ottavo capitolo per maggiori dettagli). Se invece il videoregistratore è mono, il collegamento con l'Amiga deve essere identico al collegamento che avete stabilito tra il computer e il monitor (per maggiori informazioni consultate il manuale del computer nel capitolo *Introduzione all'Amiga*).

Procedimento di registrazione

Una volta effettuati i collegamenti potete iniziare la registrazione. Per registrare tutta la sequenza evitando allo stesso tempo di mettere sul nastro anche la fase di caricamento dell'animazione, seguite attentamente queste istruzioni:

1. Fate girare Deluxe Video Maker e caricate il video che volete registrare. Selezionate **Play Video** dal menu **Project**.

2. Quando appare la sezione del controllo remoto, bloccate il video e selezionate il pulsante di azzeramento (**:0**) per riportare il video all'inizio della sequenza.

3. Selezionate il movimento per fotogrammi (single-step). Poi fate partire la riproduzione: la sequenza rimarrà subito bloccata al primo fotogramma e la linea d'intestazione sparirà.

4. Fate avanzare la sequenza di un altro fotogramma per disattivare il modo "single-step"; selezionate quindi il back gadget del controllo remoto per far sparire il riquadro del controllo stesso.

5. Piazzate il puntatore nell'angolo inferiore destro dello schermo, in modo che risulti invisibile.

6. Regolate il videoregistratore sulla più alta velocità possibile per ottenere la migliore qualità d'immagine. Premete quindi il tasto che avvia la registrazione.

7. Attendete alcuni secondi, il tempo necessario affinché le testine del registratore raggiungano una velocità costante. A questo punto premete il tasto cursore con la freccia verso destra e avviate la sequenza animata.

8. Terminata la sequenza, fermate il videoregistratore. Il video è ora registrato su cassetta.

QUALITÀ DELLE REGISTRAZIONI VIDEO

Quando riguardate le registrazioni delle vostre opere animate, non aspettatevi la stessa qualità d'immagine del video originale visualizzato su un monitor RGB. La registrazione deteriora la qualità dell'immagine per due motivi. Innanzitutto il segnale video viene convertito da segnale RGB a segnale video composito recepibile dal videoregistratore; poi il segnale composito viene registrato su cassetta, con ulteriore peggioramento della qualità originale del prodotto. Il segnale composito è purtroppo necessario, perché la maggior parte dei videoregistratori non accetta altri tipi di segnale. Tuttavia si può migliorare la qualità della registrazione acquistando un registratore ad alta fedeltà e utilizzando cassette di alta qualità. Se dovete acquistare un videoregistratore Beta, accertatevi che sia un SuperBeta, che è di qualità decisamente superiore a qualsiasi altro registratore dello stesso tipo. Se invece dovete comprare un registratore VHS, dovreste prendere in considerazione un VHS di alta qualità (HQ), che è notevolmente più affidabile.

ALTRO SOFTWARE PER L'ANIMAZIONE

Deluxe Video non è l'unico programma di animazione disponibile per l'Amiga. La Electronic Arts sta progettando di apportare dei miglioramenti a Deluxe Video stesso e la Aegis Development ha creato il programma Aegis Animator.

AEGIS ANIMATOR

Aegis animator, prodotto dalla Aegis Development, è un programma di animazione semplice ma efficace. Costa poco, è facile da usare e può creare animazioni tridimensionali fantastiche, facendo esclusivamente uso di semplici oggetti astratti. Aegis Animator viene venduto unitamente al programma grafico Aegis Images, necessario all'operatore per creare lo sfondo e gli oggetti da animare nei video.

Aegis Animator usa un particolare tipo di animazione, detta metamorfica, per la maggior parte dei suoi effetti. Nell'animazione metamorfica l'Amiga non deve calcolare il moto di ogni singola parte dell'oggetto in movimento. Il programma usa semplici figure geometriche come linee e poligoni creati con una serie di punti collegati da rette. L'Amiga, di conseguenza, deve solo calcolare la locazione degli estremi di un oggetto in movimento sullo schermo, anziché calcolare lo spostamento di ogni bit che lo costituisce. Consideriamo l'esempio di un quadrato. L'Amiga terrà conto solo della posizione dei quattro angoli. Mentre Animator sposta questa semplice figura, l'Amiga calcola le nuove locazioni degli angoli e disegna il resto dell'oggetto collegando i punti ottenuti e riempiendo la superficie della figura se necessario.

Il vantaggio dell'animazione metamorfica sta nel fatto che bastano pochi calcoli per poter muovere gli oggetti sullo schermo. Aegis Animator offre così un'ampia varietà di effetti speciali: muovere un oggetto da una locazione all'altra, variarne le dimensioni, farlo ruotare su tre assi differenti e cambiarne la forma. Altro vantaggio dell'animazione metamorfica è la maggiore fluidità di movimento, resa possibile dai tempi di calcolo molto ridotti.

Purtroppo, questo tipo di animazione consente solo di lavorare su semplici figure geometriche. Se ciò non costituisce un problema per l'animazione astratta, lo diventa certamente qualora si desideri creare oggetti come persone, animali o altri corpi di forma irregolare.

Aegis Animator offre alcune alternative all'animazione metamorfica. Per lo sfondo delle vostre animazioni potete utilizzare un'immagine a 32 colori creata con Aegis Images, Deluxe Paint o altri programmi grafici IFF standard (Aegis Animator utilizza un unico campo visivo da 5 bit-plane, mentre Deluxe Video ne usa due da 3 bit-plane; l'utente può dunque usare fino a 32 colori con il programma della Aegis Development). È inoltre possibile creare in un programma grafico oggetti che utilizzino 32 colori diversi. Aegis Animator potrà usarli per animazioni semplici: li muoverà sullo schermo ma non potrà farli ruotare su se stessi, né cambiarne la forma o le dimensioni, come fa invece con le figure geometriche semplici.

A differenza di Deluxe Video, Aegis Animator non utilizza un copione. Lascia invece che l'utente crei oggetti sullo schermo e che poi li sposti e li sottoponga ad altri effetti. L'operatore registra ogni fase del movimento e controlla tutti gli effetti, mentre il programma memorizza l'intera sequenza per poterla riprodurre in seguito. Completata una sequenza, l'operatore potrà crearne altre e fonderle poi tutte in un unico schermo speciale (detto "storyboard"). Questa tecnica, rispetto al sistema del copione, facilita la creazione di video di breve durata, ma è sconsigliabile per sequenze più lunghe, perché è alquanto difficile revisionare il video e sincronizzare gli effetti.

A differenza di Deluxe Video, Aegis Animator si limita alla pura animazione. Non esiste la possibilità di aggiungere effetti sonori o accompagnamenti musicali alle vostre creazioni, a meno che facciate girare contemporaneamente un programma musicale. Aegis Animator non dispone nemmeno di comandi che consentono di aggiungere testi all'animazione.

DISCHI LIBRARY PER DELUXE VIDEO

La Electronic Arts integra Deluxe Video con dischetti library contenenti immagini, oggetti, suoni e generatori di scene che l'utente può utilizzare nelle proprie animazioni. Il contenuto dei dischetti fornisce gli strumenti necessari per creare con estrema semplicità business graphics, grafica per la titolazione dei video... I generatori di scene consentono all'operatore di creare automaticamente scene ben precise. I dischetti vengono realizzati da artisti e programmatori professionisti con lo scopo di facilitare il vostro lavoro e migliorarne la resa finale.

L'ANIMAZIONE E LO STANDARD IFF

Dal momento che l'animazione non dispone di uno standard fisso di notazione e che quindi i diversi programmi di animazione fanno uso di tecniche completamente differenti per creare i loro effetti, non esiste uno standard IFF per i copioni dei video. Questo significa che non è possibile creare una sequenza animata su Deluxe Video e poi riprodurla utilizzando Aegis Animator o viceversa. Nonostante questo, le singole componenti dell'animazione usano lo standard IFF. Immagini, suoni e musica possono essere trasferiti da un programma a un altro che utilizzi lo standard IFF e che possa gestire le immagini create col programma originale.

ALTRO HARDWARE PER L'ANIMAZIONE

Aggiungendo nuovi elementi hardware, è possibile incrementare considerevolmente le capacità di animazione dell'Amiga. L'utente ha la possibilità di far girare i programmi molto più velocemente e di immagazzinare sequenze più lunghe di quelle memorizzabili dall'Amiga nella sua conformazione base. È inoltre possibile aggiungere fonti di animazione esterne e combinare l'animazione dell'Amiga con l'animazione esterna.

SCHEDE DI ESPANSIONE RAM

L'animazione richiede grandi quantità di memoria. Se si considera che la memorizzazione di un semplice sfondo può occupare fino a 8K di RAM e un suono campionato più di 30K, non c'è da stupirsi se i 512K (1 MB nella versione 2000) dell'Amiga si esauriscono tanto facilmente quando l'operatore crea un video con diversi sfondi, decine di oggetti e suoni campionati di accompagnamento. Per superare l'ostacolo della memoria limitata, programmi come Deluxe Video immagazzinano il maggior numero possibile di elementi dell'animazione nella RAM interna, e conservano il resto su un disco nel disk drive. Quando riproducete un video e vengono richiesti elementi conservati sul dischetto, Deluxe Video li carica cancellando dalla RAM quelli vecchi. Questa è la causa delle pause che avvengono di tanto in tanto nella riproduzione: ci vuole un po' di tempo perché Deluxe Video carichi dal dischetto i nuovi elementi dell'animazione.

Per facilitare la memorizzazione e la gestione degli elementi video, è possibile aggiungere all'Amiga fino a otto megabyte di RAM addizionale, utilizzando il connettore per l'espansione posto sul lato destro dell'unità centrale nell'A1000, il connettore sul lato sinistro nell'A500 e gli slot interni nell'A2000. Questa memoria addizionale viene chiamata RAM esterna per differenziarla dai 512K di memoria standard. È necessario che le immagini, gli oggetti e i suoni con cui l'Amiga lavora siano contenuti nella RAM standard, in quanto i coprocessori che si occupano del disegno, dell'animazione e del suono hanno accesso diretto solo alla memoria interna. È per questo motivo che tale memoria viene talvolta definita "chip memory".

Gli elementi dell'animazione che l'Amiga non utilizza subito possono essere immagazzinati nella RAM esterna. Il processore 68000 dell'Amiga può leggere direttamente da entrambe le memorie e pertanto, in caso di bisogno, può portare molto rapidamente gli elementi richiesti nella memoria interna, evitando così i ritardi causati dal caricamento da floppy disk.

Diverse aziende producono schede di RAM esterna che possono essere inserite nella porta di espansione dell'Amiga. La maggior parte di queste schede contiene 2 megabyte di RAM, una quantità di memoria sufficiente a immagazzinare il contenuto di due floppy disk. Alcune schede permettono l'ulteriore connessione di schede RAM addizionali da 2 megabyte, per un massimo consentito di 8 megabyte. Le espansioni della Commodore A501 da 512K per Amiga 500 e A2052 da 2 MB per Amiga 2000 sono due esempi di schede disponibili.

LA SCHEDA AMIGA 1300 GENLOCK

La scheda Amiga 1300 Genlock, prodotta dalla Commodore statunitense, permette di combinare la grafica e i suoni dell'Amiga con le immagini e i suoni provenienti da una fonte esterna come un televisore, un videoregistratore o un lettore di dischi laser. La scheda deve essere inserita nella porta RGB e nelle porte audio dell'unità centrale dell'Amiga; recepisce i segnali video provenienti dalla porta RGB e i segnali audio provenienti dalle porte audio. La scheda Genlock è dotata di jack che accettano gli input provenienti da fonti audiovisive esterne. La scheda ha inoltre un'uscita RGB in cui potete inserire il vostro monitor RGB, un'uscita video composita in cui potete inserire un videoregistratore e un'uscita audio a due vie per collegare l'altoparlante del monitor o un registratore, audio o video.

Utilizzando l'interruttore inserito sulla scheda Genlock, l'operatore potrà selezionare il tipo di segnale emesso dalle porte output della scheda. Se scegliete il segnale esterno, vedrete e sentirete tutto, quali che siano le fonti video e audio collegate all'Amiga. Scegliendo invece il segnale dell'Amiga, vedrete e sentirete solo la grafica e i suoni prodotti dall'Amiga. Se selezionate ambedue i segnali, vedrete e sentirete contemporaneamente i segnali esterni e quelli dell'Amiga.

Quando decidete di unire i due tipi di segnale, la scheda Genlock proietta le immagini dell'Amiga al di sopra di quelle esterne. Nelle parti dell'immagine Amiga dipinte col colore di sfondo trasparirà l'immagine della fonte esterna. I segnali audio sono combinati in modo da essere recepiti contemporaneamente.

Vediamo un esempio pratico. Se collegate l'uscita di un ricevitore televisivo all'entrata della scheda Genlock e decidete di unire il suo segnale a quello dell'Amiga, quando farete

girare il Workbench vedrete tutte le finestre, le icone e i gadget sullo sfondo in movimento costituito dalla trasmissione televisiva in onda in quel momento. Lo sfondo TV sostituisce l'ordinario schermo blu del Workbench. Allo stesso modo, l'audio emesso dal monitor sarà quello della trasmissione televisiva. E se l'Amiga vi segnala qualche errore mentre siete al lavoro, lo farà con un "beep" sovrapposto all'audio del programma TV in questione.

La scheda Genlock è particolarmente indicata per titolare i nastri videoregistrati. Collegando un videoregistratore all'entrata della scheda e un secondo videoregistratore all'uscita videocomposita, potete far girare sul primo registratore una cassetta precedentemente incisa, aggiungere grafica in sovrimpressione sull'Amiga e registrare il video titolato e commentato sul secondo videoregistratore. Considerato che gli effetti creati con Deluxe Video o Aegis Animator funzionano perfettamente anche con la scheda Genlock, non vi resta che stendere il colore di sfondo nei punti del video Amiga in cui volete che traspaiano le immagini registrate. Nei punti lasciati liberi scorreranno i titoli del vostro filmato fatto in casa.

Il Genlock 1300 è disponibile solo in versione NTSC (il sistema televisivo USA) ed è quindi destinato al mercato statunitense. La Commodore Italiana propone per il nostro Paese i modelli A2997, un'interfaccia genlock multistandard professionale, e A2998, un'interfaccia VHS/BETA.

LETTORI DI DISCHI LASER

Il lettore di dischi laser costituisce un'ottima fonte di segnali video esterni e può essere utilizzato con la scheda Genlock. Si tratta di un apparecchio che utilizza un raggio laser per la lettura dei segnali video impressi sul disco, proprio come un lettore di compact disc utilizza un laser per leggere i segnali audio. Non è possibile registrare immagini su un disco laser come su una videocassetta, ma l'impiego di un lettore laser offre numerosi vantaggi e garantisce immagini nitide e precise, suono stereo ad alta fedeltà e una varietà di effetti speciali come moviola, riproduzione all'indietro e fermo immagine. È persino possibile controllare di persona il modo in cui il lettore si integra con l'Amiga.

I lettori di dischi laser della Pioneer Video hanno conquistato il mercato statunitense. Il Pioneer CLD-900, uno degli ultimissimi modelli, legge sia dischi laser, sia compact disc; potete dunque utilizzarlo a più scopi. Questo modello è dotato di tutti gli effetti speciali propri dei lettori di dischi laser: moviola, avanzamento veloce in entrambi i sensi (con una velocità tre volte superiore a quella normale), riavvolgimento veloce in entrambe le direzioni, fermo immagine e ricerca di un singolo fotogramma. Quest'ultima caratteristica è particolarmente inte-

ressante. I dischi laser prodotti col sistema CAV (necessario per la produzione di lettori che dispongano dei diversi effetti speciali) possono immagazzinare fino a 54.000 fotogrammi, ciascuno dei quali costituisce un'immagine video completa. È possibile indicare al lettore di saltare a un certo fotogramma e visualizzarlo.

Il CLD-900 dispone di una porta input/output per connettere un'interfaccia Pioneer IU-04. La IU-04 va poi collegata alla porta seriale dell'Amiga e funge da controllo computerizzato. L'Amiga può inviare comandi attraverso la porta seriale, allo scopo di controllare il lettore laser in tutte quelle operazioni altrimenti effettuabili con i controlli propri del disco laser.

L'Amiga può essere usato in combinazione con un lettore, un'interfaccia IU-04 e una scheda Genlock, al fine di creare programmi educativi di notevole interesse. Per iniziare, potreste acquistare uno dei bellissimi dischi Space Archive della Video Vision Associates. Ogni disco Space Archive contiene sequenze filmate di allunaggi, lanci spaziali e missioni di satelliti, oltre a centinaia di immagini fisse immagazzinate una per una nei singoli fotogrammi.

L'Amiga visualizza il contenuto del disco laser sul proprio monitor attraverso la scheda Genlock. Mentre il disco gira, l'Amiga può aggiungere delle didascalie, evidenziare oggetti di particolare interesse e persino commentare i filmati con la sua voce sintetizzata. In qualsiasi punto del disco, l'Amiga, grazie alla IU-04, può bloccare l'immagine e chiedere allo spettatore se desidera per caso passare a un'altra sezione del disco; magari potrebbe anche porre domande per un breve quiz. L'Amiga può infine passare a un'altra sezione del disco, a seconda delle indicazioni dell'operatore.

Attualmente non sono presenti in commercio programmi che lavorino con i dischi laser e la scheda Genlock in questo modo, ma l'hardware e il disco sono in vendita: potete utilizzare l'Amiga BASIC per creare un programma che faccia tutto quello che abbiamo appena descritto.

In questo capitolo abbiamo illustrato alcune tecniche per creare dei video con l'Amiga per mezzo di Deluxe Video. Abbiamo anche fatto una panoramica di altri elementi hardware e programmi di animazione disponibili per l'Amiga. Se volete informazioni più dettagliate sull'animazione con l'Amiga, voltate pagina: i prossimi due capitoli descrivono i comandi in BASIC e i procedimenti di base essenziali per la creazione di sequenze animate.

DODICESIMO CAPITOLO L'ANIMAZIONE E L'AMIGA BASIC: CREAZIONE DI OGGETTI IN MOVIMENTO

L'Amiga BASIC contiene una grande varietà di comandi e di funzioni che permettono di muovere gli oggetti sullo schermo, in diverse direzioni e a varie velocità. Si può far passare un oggetto sull'altro, farlo sparire e riapparire. L'Amiga BASIC segue i movimenti di tutti gli oggetti sullo schermo e controlla che non urtino l'uno contro l'altro e non vadano a sbattere contro i margini della finestra nella quale si muovono. L'Amiga BASIC fornisce tutti gli strumenti per creare interessanti animazioni.

La prima tappa consiste nel disegnare l'oggetto da animare con l'Object Editor, un programma contenuto nel disco dell'Amiga BASIC. L'Object Editor può salvare gli oggetti su disco per riutilizzarli successivamente in qualsiasi programma BASIC. Quando si scrive un programma, ci si può avvalere dei comandi BASIC di animazione per riportare in memoria gli oggetti conservati su disco, posizionarli nella finestra output, fissarne la velocità, iniziare a muoverli, fermarli e rimuoverli dallo schermo. È possibile inserire anche altri comandi che indicano la velocità e la posizione delle figure sullo schermo e controllano la sequenza delle loro collisioni.

Questo capitolo spiega come utilizzare il programma Object Editor per creare gli oggetti; sperimenterete anche alcuni comandi BASIC per l'animazione che posizionano gli oggetti sullo schermo e li fanno muovere. Nel tredicesimo capitolo imparerete a controllare il movimento degli oggetti, a farli passare uno sopra l'altro e a controllarne le collisioni: tutto questo per permettervi di creare animazioni veramente realistiche.

COME USARE L'OBJECT EDITOR

Il primo passo per creare animazioni con il BASIC consiste nel disegnare gli oggetti con l'Object Editor. Se osservate il contenuto del disco dell'Amiga BASIC, vedrete un cassetto chiamato **BasicDemos**. In esso è contenuta una serie di programmi BASIC che dimostrano le capacità dell'Amiga BASIC. Uno di questi programmi è l'Object Editor, indicato come **ObjEdit**.

Per utilizzare l'Object Editor iniziate come con qualsiasi altro programma BASIC: puntate l'icona e schiacciate due volte il tasto del mouse, oppure selezionate **Open** dal menu **Project** del BASIC. Il settimo capitolo del manuale dell'Amiga BASIC, "Creazione di immagini animate", spiega come servirsi

dell'Object Editor. È bene leggere il capitolo per vedere come selezionare i colori, utilizzare i comandi di animazione e salvare e richiamare gli oggetti da disco.

COME MODIFICARE L'OBJECT EDITOR

L'Object Editor presente sul disco dell'Amiga BASIC è stato creato appositamente per lavorare su un Amiga con 256K (il primo modello di A1000), e si limita a creare oggetti con due soli bit-plane (quattro colori), su uno schermo che utilizza il modo di risoluzione 2. Dal momento che l'Amiga può utilizzare oggetti con profondità e risoluzioni diverse, è meglio modificare l'Object Editor adattandolo a un Amiga da 512K, in modo da poter sfruttare tutte le risoluzioni e profondità offerte dalla macchina. Fortunatamente è semplice adattare il programma, basta apportare qualche lieve modifica.

Innanzitutto aprite l'Object Editor. Se sta girando, fermatelo, aprite la finestra List e ingranditela in modo da vedere le linee del programma in tutta la loro lunghezza. Se fate scorrere il listato, vedrete che è piuttosto lungo, ma diviso in brevi sezioni, la maggior parte delle quali inizia con un'etichetta (label) e termina con una linea vuota.

Per modificare l'Object Editor, sostituirete alcune sezioni e ne correggerete altre. Per visualizzare solo una sezione particolare, selezionate la finestra output e immettete **list** seguito da uno spazio e dal nome della sezione che vi interessa; la finestra List listerà il programma a partire dalla prima linea della sezione richiesta. È possibile limitarsi a modificare una sezione oppure sostituirla completamente.

Per modificare l'Object Editor, seguite attentamente le seguenti istruzioni:

1. Contate le linee del listato partendo dall'inizio del programma. Sostituite dalla linea 30 alla 38 (che iniziano con *scrn = 1* e terminano con *WINDOW 1,,(0,0) - (WinX,WinY) ,31,scrn)* con questa nuova sottosezione:

```
SetScreen:
    INPUT ''Resolution Mode (1-4)?'', res
    INPUT ''Depth (Mode 1: 1-5, mode 2 & 3: 1-4, mode 4: 1-3)?'', depth
    IF depth < 1 THEN depth = 1
    IF res = 4 THEN
         wide = 640: high = 400
         IF depth > 3 THEN depth = 3
    ELSEIF res = 3 THEN
         wide = 320: high = 400
         IF depth > 4 THEN depth = 4
    ELSEIF res = 2 THEN
         wide = 640: high = 200
```

(segue)

```
        IF depth > 4 THEN depth = 4
    ELSE
        wide = 320: high = 200
        IF depth > 5 THEN depth = 5
    END IF
    WinX = wide - 9: WinY = high - 15
    scrn = 1
    SCREEN scrn, wide, high, depth, res
    WINDOW 2, ''Object Editor'', (0,0) - (WinX,WinY), 0, scrn
```

2. Listate la sottosezione *StartOver:*. Alla fine della sottosezione inserite la seguente linea fra quelle di *MENU RESET* e *CLS*:

```
SCREEN CLOSE scrn
```

3. Listate la subroutine *InitConstant:*. Sostituite la settima linea *StatusLine = 20* con:

```
StatusLine = 23
```

4. Listate la subroutine *InitFile:*. Modificate la terza linea da *IF Depth = 2 THEN* in:

```
IF Depth = 2 AND res = 1 THEN
```

5. Listate la subroutine *PrintColorBar:*. Sostituitela interamente con la seguente versione modificata:

```
PrintColorBar:
    COLOR CurrentColor
    LOCATE 19, 1: PRINT ''Color:'';
    ColorBar = WINDOW(5) - 10
    COLOR 1
    x = 70: newbar = 0: barcount = 0: barend = 640
    FOR i = 0 TO maxColor
        z = newbar + ColorBar
        LINE (x,z) - (x + 20,z + 10), i, bf
        LINE (x,z) - (x + 20,z + 10), i, b
        x = x + 20
        IF x + 20 > WINDOW(2) THEN
            newbar = newbar + 10
            barcount = (x - 70) / 20
            barend = x: x = 70
        END IF
    NEXT i
    RETURN
```

6. Listate la subroutine *CheckColor:*. Sostituitela interamente con la seguente versione modificata:

```
CheckColor:
    IF CurrentY < ColorBar THEN RETURN
    IF CurrentY > ColorBar + newbar + 10 THEN RETURN
    IF CurrentX < 70 THEN RETURN
    IF CurrentX > barend THEN RETURN
    i = INT((CurrentX - 70) / 20)
    i = i + INT((CurrentY - ColorBar) / 10) * barcount
    IF i > maxColor THEN RETURN
    CurrentColor = i
    GOSUB PrintColorBar
    RETURN
```

7. Salvate l'Object Editor così modificato con un nuovo nome, ad esempio **NewEdit**.

8. Assicuratevi che la fonte del testo di default per il BASIC sia fissata sulle 80 colonne. Il testo del nuovo Object Editor non si adatta allo schermo se si usa la fonte a 60 colonne. Se non avete le 80 colonne già pronte, uscite dal BASIC, selezionate Preferences nel Workbench per cambiare la fonte e ricaricate infine il BASIC.

COME DISEGNARE UN OGGETTO

Il nuovo Object Editor può essere utilizzato per creare sprite e bob (di cui abbiamo parlato nel decimo capitolo). Fate girare il programma: all'inizio sarà richiesto un modo di risoluzione, poi la profondità (depth) in bit-plane. Se volete disegnare uno sprite, inserite la risoluzione 1 e la profondità 2 (probabilmente dovrete prima selezionare la finestra per attivarla).

Se volete disegnare un bob, scegliete un modo di risoluzione che corrisponda a quello che volete utilizzare con il vostro programma di animazione. Scegliete la profondità dello schermo che vi permette di usare tutti i colori di cui avete bisogno. Ricordate che le diverse profondità in bit-plane vi offrono un numero diverso di colori con cui lavorare: 1 bit-plane = 2 colori; 2 bit-plane = 4 colori; 3 bit-plane = 8 colori; 4 bit-plane = 16 colori; 5 bit-plane = 32 colori.

Dopo aver scelto il modo di risoluzione e la profondità, appare lo screen editor (editor di schermo), a meno che non abbiate scelto una risoluzione di modo 1 e una profondità di 2 bit-plane, nel qual caso vi verrà chiesto se volete creare uno sprite o un bob. Dopo aver scelto apparirà lo screen editor (riportato nella Figura 12-1) e potrete iniziare a disegnare l'oggetto.

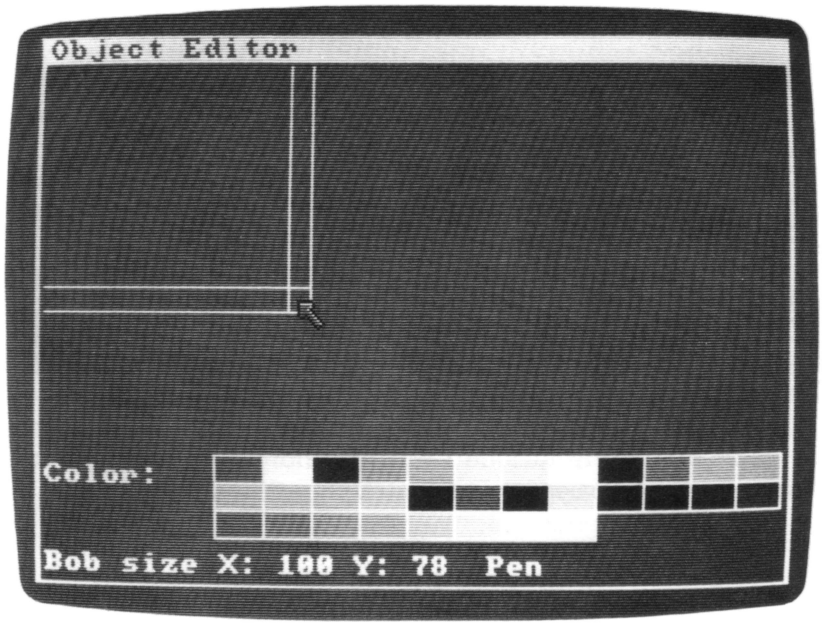


Figura 12-1.

Lo schermo editor dell'Object Editor, selezionato sul modo di risoluzione 1 con 5 bit-plane.

Creazione di uno sprite

Quando disegnate uno sprite, l'area di lavoro dell'Object Editor ha un'ampiezza fissata, che limita la larghezza massima dello sprite stesso. Le dimensioni sono infatti stabilite

dall'hardware che disegna gli sprite sullo schermo. Comunque l'altezza dell'area può essere modificata a seconda dello sprite che volete creare.

Dopo aver fissato l'altezza, si ha una scelta di quattro colori con cui operare. Ricordate che la prima tinta nella lista dei colori dell'Object Editor, quella dello sfondo, risulta trasparente quando lo sprite appare nella finestra. Ogni parte che avete colorato con la tinta dello sfondo (o non avete colorato, visto che tutti i pixel sono del colore dello sfondo) permette di vedere lo spazio e gli oggetti sottostanti allo sprite.

Dopo aver finito di disegnare lo sprite, potete salvarlo su disco utilizzando il comando **Save** del menu **File**. Per rendere le cose più facili in seguito, è meglio aggiungere **.spr** alla fine del nome di ogni sprite. Ad esempio, potete chiamare uno sprite a forma di matita **Matita.spr**. Successivamente, quando guarderete i nomi di tutti gli oggetti che avete creato e memorizzato su disco, potrete identificarne gli sprite leggendo i suffissi.

Ora disegnate uno sprite a forma di ape e memorizzatelo col nome di **Ape.spr**. Lo utilizzerete negli esempi di programma del tredicesimo capitolo.

Creazione di un bob

Per disegnare un bob, si può usare una qualsiasi delle quattro risoluzioni e un numero di colori qualsiasi (fino a 32), stabilito a seconda della profondità fissata. Il bob può essere di dimensioni svariate. È possibile ampliare l'area in cui lavorate, sia in senso orizzontale, sia in altezza. L'Object Editor impedisce automaticamente di ampliare l'area tanto da creare un bob troppo grande per la memoria.

I colori visualizzati nella parte inferiore dello schermo sono i colori preselezionati contenuti nei registri dello schermo. Dopo aver scelto le tinte, ricordate che, quando il bob appare sullo schermo, il primo colore (quello dello sfondo) è trasparente, proprio come quando utilizzate il colore dello sfondo per uno sprite.

Dopo aver salvato il bob su disco, bisogna aggiungere un suffisso al nome, per indicare la risoluzione e il numero di bit-plane che vengono impiegati. Ad esempio, se create un bob che ha la forma di un'automobile, con una risoluzione di modo 2 e 3 bit-plane, potete chiamarlo **Auto.2bob3**. Potete anche creare un bob a forma di sandwich, con una risoluzione di modo 1 e 5 bit-plane, e chiamarlo **Panino.1bob5**.

Prima di smettere di realizzare oggetti con l'Object Editor, disegnate diversi bob con modo di risoluzione 1; ve ne servirete nei successivi programmini dimostrativi. Disegnate un bob con un bit-plane soltanto e salvatelo come **Uccello.1bob1**, un bob con tre bit-plane e memorizzatelo come **Fiore.1bob3**, e uno con 5 bit-plane e salvatelo come **Robot.1bob 5**. Non fateli troppo grandi! Quando avete terminato, selezionate **Quit** per uscire dal programma.

CREAZIONE DEL CAMPO VISIVO

Dopo aver disegnato e salvato alcuni bob e sprite, il primo passo per creare un programma di animazione in BASIC che li utilizzi consiste nel realizzare un campo visivo, uno sfondo su cui gli oggetti si muovano. Innanzitutto bisogna creare uno schermo e una finestra output con i comandi SCREEN e WINDOW che avete visto nel quarto capitolo. Se volete un campo visivo che adoperi altri colori oltre a quello dello sfondo, potete servirvi di alcuni dei comandi BASIC grafici e di stampa che consentono di creare immagini nella finestra output. Quando inserite sullo schermo gli oggetti utilizzando i comandi di animazione BASIC, tutto ciò che avrete inserito nel campo visivo farà parte dello sfondo.

RISOLUZIONE DELLO SCHERMO

Quando scegliete la risoluzione dello schermo con il comando SCREEN, cercate di farla corrispondere a quella dei bob che volete utilizzare. Potete anche inserire un bob creato con una certa risoluzione in uno schermo con risoluzione diversa, ma le proporzioni della figura saranno diverse dall'originale a causa delle modifiche subite dai pixel che la compongono. Ad esempio, un bob di risoluzione 1 sarà largo la metà in uno schermo a risoluzione 2, perché i pixel di quest'ultimo sono più stretti. La Figura 12-2 mostra un bob creato con il modo di risoluzione 1 e inserito in schermi a risoluzione 1, 2, 3 e 4.

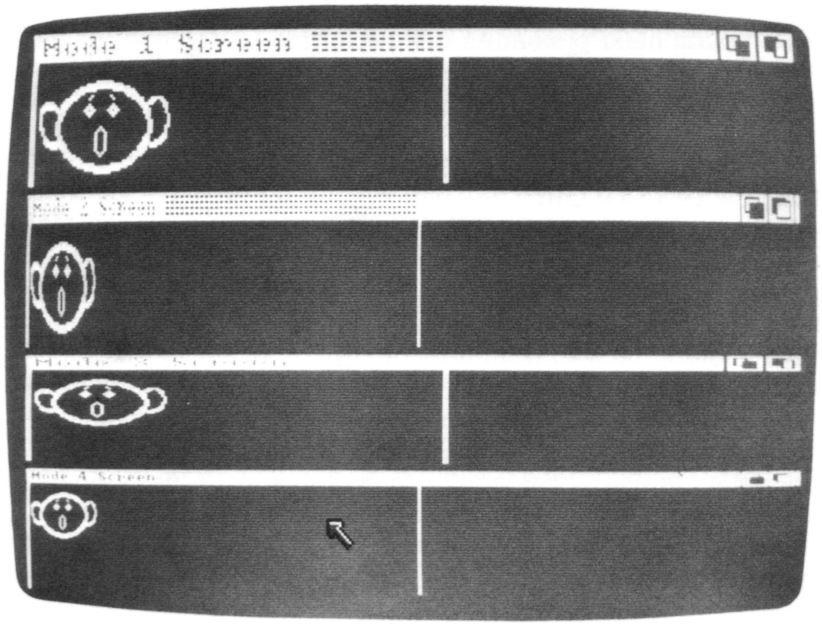


Figura 12-2.

Un bob a risoluzione 1 visualizzato su schermi a risoluzione 1, 2, 3 e 4.

Gli sprite, invece, sono sempre uguali, indipendentemente dalla risoluzione dello schermo in cui vengono inseriti. Questo perché uno sprite non è parte integrante dello schermo, bensì viene visualizzato dall'hardware dell'Amiga separatamente, utilizzando i pixel del modo 1.

PROFONDITÀ DELLO SCHERMO

Per sfruttare al meglio tutti i colori che avete scelto per un bob, dovete creare uno schermo che possa contenere l'intera gamma di colori del bob a cui avete assegnato il maggior numero di bit-plane. Ad esempio, se volete utilizzare due bob, uno di due e l'altro di tre bit-plane, dovete creare uno schermo di almeno tre bit-plane per poter sistemare entrambi.

Volendo, potete anche mettere un bob su uno schermo con un numero insufficiente di bit-plane, ma non potrete usare tutti i colori e dovrete limitarvi alla quantità consentita dallo schermo stesso. Ad esempio, se mettete un bob di 5 bit-plane in uno schermo di 2 bit-plane, esso sarà limitato ai 2 bit-plane e ai quattro colori dello schermo. Tutti i pixel che utilizzano una tinta non compresa nello schermo appariranno con il colore

dello sfondo e si vedranno solo le parti colorate con tinte contemplate nella profondità dello schermo.

Gli sprite, invece, utilizzano registri di colore propri e la profondità dello schermo sul quale appaiono non ha alcun effetto.

SCELTA DEI COLORI

Quando visualizzate i bob e gli sprite sullo schermo, quest'ultimo utilizza per il bob i colori memorizzati nei propri registri. Queste tonalità possono essere diverse da quelle usate dall'utente per il disegno originale, perché l'Object Editor non salva i colori, ma solo il numero di registro di tutti i pixel utilizzati per disegnare il bob, partendo da 0 per il colore dello sfondo e procedendo verso destra all'interno della gamma cromatica dell'Object Editor. Se i colori vanno oltre la prima riga della gamma, il conteggio riprende a sinistra nella riga successiva e procede verso destra. Ad esempio, uno schermo Object Editor a risoluzione 1 con 32 colori, viene numerato come mostra la Figura 12-3.

Figura 12-3.

I registri di colore utilizzati nello schermo dell'Object Editor, a risoluzione 1 e con 5 bitplane.

0	1	2	3	4	5	6	7	8	9	10	11
12	13	14	15	16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31				

Per utilizzare anche nell'Object Editor i colori con cui lavorerete sullo schermo, potete modificare ulteriormente il programma. Listate la sezione del programma denominata *SetScreen* e inserite all'inizio della sezione gli stessi comandi PALETTE che utilizzerete per fissare i nuovi colori dello schermo (i comandi **Cut** e **Paste** nel menu **Edit** funzionano molto bene a questo scopo). Quando fate girare l'Object Editor, quest'ultimo utilizzerà i colori appena fissati.

Gli sprite si servono di uno schema di colori diverso dai bob. Quando l'Object Editor salva uno sprite su disco, memorizza automaticamente il suo primo colore (secondo del Workbench) come turchese, il secondo (terzo del Workbench) come nero e il terzo (quarto del Workbench) come arancione. Successivamente, quando richiamate lo sprite sullo schermo in un programma Amiga BASIC, esso verrà visualizzato con quei colori.

L'Amiga utilizza i registri dal 17 al 19 per memorizzare i

colori dei primi due sprite sullo schermo, quelli da 21 a 23 per i due sprite successivi; la terza coppia di sprite occupa i registri dal 25 al 27 e la quarta quelli dal 29 al 31. Quando il BASIC carica uno sprite da disco, assegna al registro che lo sprite adopererà i colori memorizzati con lo sprite stesso, cambiando le tinte che potrebbero essere presenti. Se state utilizzando uno schermo con 16 colori o meno, ciò non ha conseguenze sulle tinte dello schermo. Se state utilizzando uno schermo a 32 colori, le tinte sul monitor che impiegano lo stesso registro dello sprite cambieranno non appena questo apparirà.

Se volete creare uno sprite con colori diversi dal turchese, nero e arancione, potete listare la sezione dell'Object Editor chiamata *SaveFile:* e cambiare le linee 20, 21 e 22. Queste linee assegnano al file dello sprite tre valori RGB esadecimali, uno per ogni colore dello sprite. Il valore tra parentesi in ogni riga, dopo MKI$, contiene un numero esadecimale a tre cifre (precedute da un "&H") che specifica la quantità di rosso, verde e blu contenuta nel colore dello sprite. La prima cifra a sinistra è il valore del rosso, la seconda quello del verde e la terza quello del blu. Ogni cifra rappresenta la proporzione o l'intensità del relativo colore su una scala da 0 a 15 (in cifre esadecimali, da 0 a F). Così, un valore di FFF rappresenta il bianco (tutti e tre i colori fondamentali alla massima intensità), mentre il valore 000 corrisponde al nero (tutti i colori "spenti").

DIMENSIONI DELLA FINESTRA

Qualsiasi oggetto o campo visivo che create deve essere visualizzato in una finestra. Per la finestra output sono ammesse una risoluzione e una grandezza qualsiasi, ma per ottenere i migliori risultati possibili, è meglio avere a disposizione parecchio spazio per il movimento degli oggetti. La finestra di tipo 0 (a tutto schermo, senza linea d'intestazione, senza gadget e senza "refreshing") è quella che offre più spazio per lavorare. Inoltre non può essere modificata né spostata, operazioni che possono interferire con le animazioni in corso.

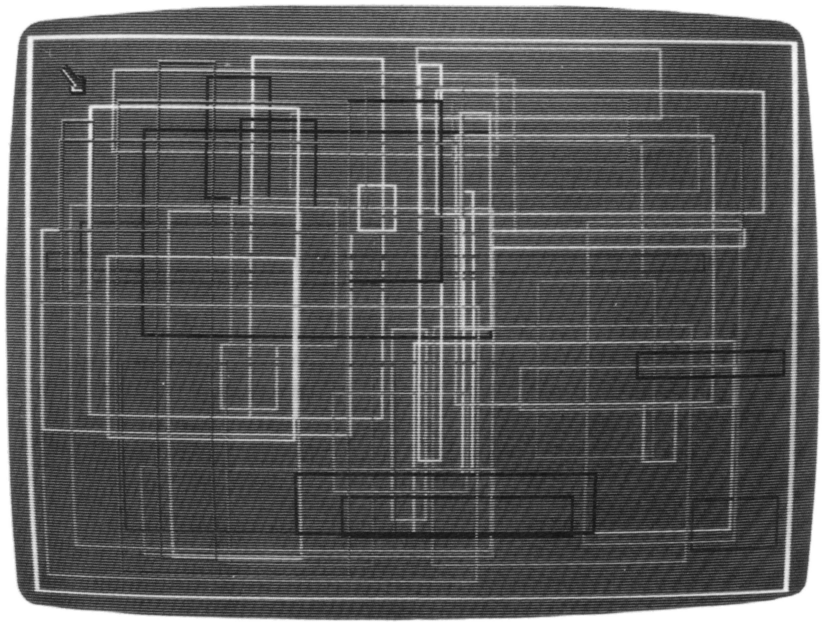
I comandi che seguono definiscono un campo visivo per i comandi di animazione che utilizzerete più avanti nel capitolo Il comando SCREEN crea uno schermo a risoluzione 1 e profondo 5 bit-plane. Il comando WINDOW crea una finestra di tipo 0 senza linea d'intestazione, per usarla interamente come campo visivo. Il ciclo FOR...NEXT disegna 50 riquadri colorati di grandezza casuale per dare vivacità allo sfondo.

```
MakePlayfield:
    SCREEN 1, 320, 200, 5, 1
    WINDOW 2, , , 0, 1
    FOR i = 1 TO 50
        x1 = INT(WINDOW(2) * RND)
        x2 = INT(WINDOW(2) * RND)
        y1 = INT(WINDOW(3) * RND)
        y2 = INT(WINDOW(3) * RND)
        LINE (x1,y1) - (x2,y2), INT(32 * RND), B
    NEXT i
```

Questa sezione di programma, chiamata *MakePlayfield*, crea un campo visivo per sprite e bob realizzati in precedenza con l'Object Editor e predisposti per il modo di risoluzione 1. La Figura 12-4 mostra il campo visivo in questione. Utilizzerete la sezione di programma appena vista in esempi successivi.

Figura 12-4.

Un campo visivo in modo 1.



CARICAMENTO, POSIZIONAMENTO E MOVIMENTO DEGLI OGGETTI NEL CAMPO VISIVO

Dopo aver creato uno schermo e aver aperto una finestra per il campo visivo, è possibile inserire gli oggetti e muoverli con una serie di comandi di animazione dell'Amiga BASIC. Questi comandi BASIC operano in cooperazione ed è consigliabile utilizzarli con ordine per ottenere buoni risultati.

1. Portate in memoria un oggetto precedentemente salvato sul disco con l'Object Editor con il comando OBJECT.SHAPE.

2. Specificate dove volete posizionare l'oggetto nel campo visivo, con i comandi OBJECT.X e OBJECT.Y.

3. Utilizzate il comando OBJECT.ON per far apparire l'oggetto nella posizione specificata.

4. Stabilite la velocità e la direzione di un movimento dell'oggetto con i comandi OBJECT.VX e OBJECT.VY.

5. Iniziate a muovere gli oggetti con il comando OBJECT.START.

6. Aumentate o diminuite la velocità di un oggetto con i comandi OBJECT.AX e OBJECT.AY.

7. Fermate il movimento di un oggetto con il comando OBJECT.STOP.

8. Fate scomparire un oggetto dal campo visivo con il comando OBJECT.OFF.

9. Cancellate un oggetto dalla memoria dell'Amiga con il comando OBJECT.CLOSE.

I comandi non devono necessariamente susseguirsi in questo ordine; ad esempio si può utilizzare OBJECT.STOP prima di OBJECT.AX e OBJECT.AY, ma in genere l'ordine è importante. Se cercate di usare uno qualsiasi dei comandi OBJECT prima di immettere OBJECT.SHAPE, non otterrete alcun risultato. Allo stesso modo, è importante stabilire la velocità di un oggetto prima di OBJECT.START, o iniziare a muoverlo prima di cercare di fermarlo con il comando OBJECT.STOP.

Studiando i comandi di animazione presenti in questa sezio-

ne, vedrete che i programmi dimostrativi per ogni comando non hanno alcun effetto, poiché ogni singolo comando deve lavorare con gli altri per creare gli oggetti e muoverli. Per far funzionare gli esempi, dovete unirli ad altre sezioni di programma e creare un listato completo che possa girare. Le istruzioni che precedono ogni esempio vi diranno come cambiare le sezioni per realizzare un programma funzionante.

IL COMANDO OBJECT.SHAPE

Il comando OBJECT.SHAPE svolge due funzioni: crea un oggetto nella memoria dell'Amiga grazie alle informazioni memorizzate su disco dall'Object Editor, e duplica un oggetto già in memoria. Verrà spiegato più avanti come duplicare gli oggetti. Per richiamare un oggetto in memoria, OBJECT.SHAPE utilizza la seguente sintassi:

```
OBJECT.SHAPE numero di ID dell'oggetto, stringa di definizione
dell'oggetto
```

Il numero di ID dell'oggetto può essere un numero intero compreso fra 1 e il numero dell'ultimo oggetto che trova posto in memoria. La stringa di definizione è un'espressione che contiene i dati necessari alla creazione dell'oggetto.

Quando il BASIC esegue OBJECT.SHAPE, crea un oggetto basandosi sui dati della stringa di definizione e lo assegna al numero di ID specificato. È possibile utilizzare il numero di ID con gli altri comandi OBJECT per identificare l'oggetto. Se usate due comandi OBJECT.SHAPE con lo stesso numero di ID, il secondo comando elimina dalla memoria l'oggetto creato col primo comando e ne crea uno nuovo con lo stesso numero di ID.

Quando salvate un oggetto su disco con l'Object Editor, questo lo memorizza con la sequenza richiesta dalla stringa di definizione. Ciò facilita l'operazione di trasferimento di un oggetto nel programma BASIC; non dovete preoccuparvi di definirlo, perché l'Object Editor lo ha già fatto per voi.

Per portare nella stringa di definizione i dati memorizzati su disco, dovete prima aprire il file creato dall'Object Editor con un comando OPEN, quindi assegnare il contenuto del file a una stringa con un comando INPUT$, e infine chiudere il file con un comando CLOSE. L'esempio che segue apre il file per un bob chiamato *Sigaro.1bob3*, crea l'oggetto numero 1 con i dati di quel bob e poi chiude il file.

```
OPEN ''Sigaro.1bob3'' FOR INPUT AS 1
OBJECT.SHAPE 1, INPUT$(LOF(1),1)
CLOSE 1
```

Nella prima linea, il comando OPEN apre il canale di comunicazione numero 1 tra il file e la memoria. Nella seconda linea, OBJECT.SHAPE crea un oggetto numerandolo con 1, con una stringa fornita dal comando INPUT$. Questo comando crea la stringa leggendo tutto il file trovato grazie a LOF(1) e aperto sul canale di comunicazione 1. La terza linea chiude il canale di comunicazione. Se volete avere maggiori informazioni su OPEN, CLOSE, INPUT$ e LOF(), consultate il manuale dell'Amiga BASIC.

DATI DELL'OGGETTO MEMORIZZATI NEI CASSETTI

Quando il BASIC apre il file dell'oggetto, si aspetta di trovare l'oggetto nello stesso cassetto del Workbench in cui è memorizzato il programma che sta girando in quel momento. Se avete memorizzato il file in un altro cassetto, il BASIC non lo troverà e vi segnalerà l'errore con il messaggio *File not found*. Ad esempio, immaginate di creare un cassetto nel Workbench chiamato **Animation**. All'interno di questo cassetto createne altri due, **Programs** e **Objects**. Memorizzate tutti i vostri programmi di animazione nel cassetto Programs e tutti i file creati con l'Object Editor nel cassetto Objects. Quando fate girare un programma conservato nel cassetto Programs, il comando OPEN cercherà il file dell'oggetto nel cassetto Programs e non lo troverà perché è stato immagazzinato nel cassetto Objects.

Per fare in modo che il programma funzioni indipendentemente da dove lo memorizzate, potete specificare i cassetti nel comando OPEN. Nell'esempio precedente potete usare il comando

```
OPEN '':Animation/Objects/Sigaro.1bob3'' FOR INPUT AS 1
```

al posto del comando OPEN precedente. Il nome del file inserito nel nuovo comando e definito "pathname", è preceduto dai due punti, che chiedono a OPEN di iniziare la sua ricerca dallo stesso disco che l'utente sta utilizzando per far girare il BASIC. Poi viene ordinato a OPEN di cercare nel cassetto Animation, quindi di trovare e ispezionare il cassetto Objects, in cui dovrebbe finalmente trovare il file "Sigaro.1bob3". Utilizzando un "pathname", potete memorizzare gli oggetti in un cassetto diverso da quello del programma, far girare il programma e far sì che il comando OPEN trovi comunque il file dell'oggetto in questione.

Per creare il "pathname" di un oggetto, iniziate sempre con i due punti o con il nome del disk drive (ad esempio, **df1**:); inserite poi i nomi dei cassetti nei quali avete memorizzato gli

oggetti, partendo dal primo cassetto che appare nel Workbench fino all'ultimo che contiene i vostri oggetti. Infine scrivete il nome del file dell'oggetto. Separate i nomi del cassetto e il nome del file con delle barre.

Non dimenticate che gli oggetti creati dall'Object Editor saranno memorizzati automaticamente nel cassetto che contiene l'Object Editor, a meno che non specifichiate qualcosa di diverso con un "pathname". Per evitare confusioni dovreste sempre copiare i vostri oggetti nel cassetto da cui intendete ripescarli subito dopo essere usciti dall'Object Editor.

Per creare gli oggetti per il vostro programma dimostrativo, aggiungete all'esempio precedente la seguente sezione di listato. I nomi dei file danno per scontato il fatto che gli oggetti siano nello stesso cassetto del programma. Questa sezione del programma, chiamata *MakeObjects*, crea tre oggetti nella memoria dell'Amiga numerandoli 1, 2, 3:

```
MakeObjects:
    OPEN ''Fiore.1bob3'' FOR INPUT AS 1
    OBJECT.SHAPE 1, INPUT$(LOF(1), 1)
    CLOSE 1
    OPEN ''Robot.1bob5'' FOR INPUT AS 1
    OBJECT.SHAPE 2, INPUT$(LOF(1), 1)
    CLOSE 1
    OPEN ''Ape.spr'' FOR INPUT AS 1
    OBJECT.SHAPE 3, INPUT$(LOF(1), 1)
    CLOSE 1
```

I COMANDI OBJECT.X E OBJECT.Y

Dopo aver creato un oggetto in memoria con il comando OBJECT.SHAPE, è possibile inserirlo sullo sfondo utilizzando i comandi OBJECT.X e OBJECT.Y, con la sintassi seguente:

```
OBJECT.X numero di ID dell'oggetto, coordinata x

OBJECT.Y numero di ID dell'oggetto, coordinata y
```

Il numero di ID dell'oggetto è un qualsiasi numero di ID che l'utente ha usato in precedenza per creare un oggetto con OBJECT.SHAPE. Le coordinate x e y possono essere numeri interi compresi fra -32768 e +32767.

Per capire come questi comandi posizionano un oggetto all'interno del campo visivo, è bene sapere come si regolano con i margini di un oggetto. Quando create una figura con l'Object Editor, anche se piccola e posta al centro dell'area di

lavoro, in realtà si tratta di un oggetto con margini rettangolari grandi quanto l'area di lavoro dell'Object Editor. È probabile che i margini siano invisibili; forse avete lasciato intorno all'oggetto dello spazio di sfondo che non appare nel campo visivo. Quando OBJECT.X e OBJECT.Y posizionano un oggetto, utilizzano come punto di riferimento l'angolo superiore sinistro del margine, come mostra la Figura 12-5.

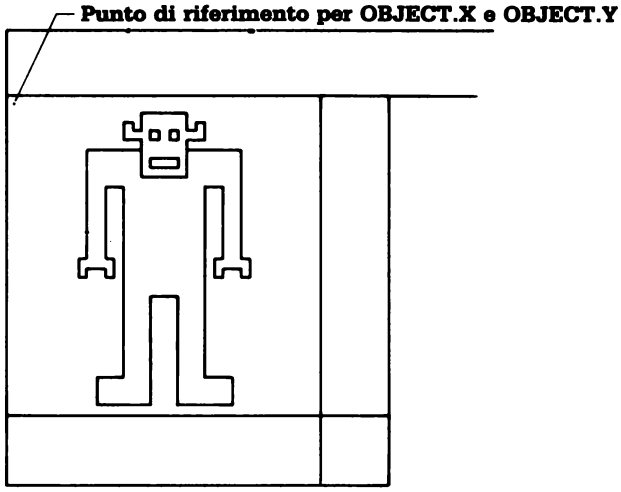


Figura 12-5.

I comandi OBJECT.X e OBJECT.Y posizionano l'oggetto utilizzando come riferimento l'angolo superiore sinistro.

Dovreste quindi cercare di creare oggetti il più vicino possibile all'angolo superiore sinistro del campo. Ciò facilita e rende più preciso il posizionamento di un oggetto, soprattutto se è molto più piccolo dell'area di lavoro. Non dovete temere che i margini invisibili interferiscano col modo in cui gli oggetti si toccano: l'Amiga BASIC utilizza i margini solo per posizionare le figure e li ignora quando esse si toccano, nel qual caso tiene esclusivamente conto dei pixel che non fanno parte dello sfondo.

Per posizionare un oggetto con OBJECT.X e OBJECT.Y, rintracciate l'indirizzo del pixel nel quale volete collocare l'angolo superiore sinistro dell'oggetto. Inserite OBJECT.X seguito dal numero di ID dell'oggetto e dalla coordinata x (orizzontale) del pixel. Immettete poi OBJECT.Y con il numero di ID dell'oggetto

e la coordinata y (verticale). Ad esempio, per collocare l'angolo superiore sinistro dell'oggetto 1 sul pixel (34,50), usate i seguenti comandi:

```
OBJECT.X 1, 34
OBJECT.Y 1, 50
```

Se non specificate le coordinate x e y con i comandi OBJECT.X e OBJECT.Y, il BASIC posiziona l'oggetto al pixel (0,0).

È anche possibile collocare un oggetto completamente all'esterno dei margini della finestra che state utilizzando come campo visivo. Se le coordinate x e y sono superiori ai margini della finestra, l'oggetto sarà creato al di fuori dell'area visualizzata. Ad esempio:

```
OBJECT.X 1, 1028
OBJECT.Y 1, 50
```

posizionerà l'oggetto oltre il lato destro della finestra, anche se questa è a tutto schermo e a risoluzione 4, cioè con 640 pixel in senso orizzontale.

Per posizionare i tre oggetti che avete creato nel paragrafo precedente, aggiungete questa sezione al programma:

```
PlaceObjects:
    OBJECT.X 1, 0: OBJECT.Y 1, 0
    OBJECT.X 2, 150: OBJECT.Y 2, 0
    OBJECT.X 3, 0: OBJECT.Y 3, 150
```

Queste linee di programma collocano il primo oggetto nell'angolo superiore sinistro del campo visivo, il secondo in alto al centro e il terzo nell'angolo inferiore sinistro. Non potrete però vederli finché non saranno "accesi" dal comando OBJECT.ON.

IL COMANDO OBJECT.ON

Con i comandi OBJECT.X e OBJECT.Y l'oggetto non appare sul campo visivo. È il comando OBJECT.ON che lo visualizza nel punto in cui è stato posizionato. Il comando usa la seguente sintassi:

OBJECT.ON *numero di ID dell'oggetto, numero di ID dell'oggetto, ...*

I numeri di ID indicano gli oggetti creati in precedenza nel programma con OBJECT.SHAPE. Il comando OBJECT.ON può

essere seguito da tutti i numeri che trovano posto su una riga, purché separati da virgole. Si può inserire un solo numero o anche nessuno, e in questo caso appariranno sullo schermo, nella locazione stabilita da OBJECT.X e OBJECT.Y, tutti gli oggetti creati con OBJECT.SHAPE. Se non avete specificato la locazione, tutte le figure saranno visualizzate al pixel (0,0), nell'angolo superiore sinistro dello schermo.

Se OBJECT.ON è seguito da più numeri di ID, appariranno sul campo visivo solo gli oggetti specificati da tali numeri, mentre gli altri resteranno invisibili. Ad esempio, il comando

```
OBJECT.ON 3, 4
```

visualizzerà sullo schermo gli oggetti 3 e 4.

Per far apparire tutti gli oggetti dell'esempio precedente, aggiungete questa linea alla fine della sezione denominata *PlaceObjects:*.

```
OBJECT.ON
```

Per vedere le figure, provate a far girare il programma. Dovrebbero apparire, secondo le coordinate immesse in precedenza, nelle posizioni mostrate dalla Figura 12-6.

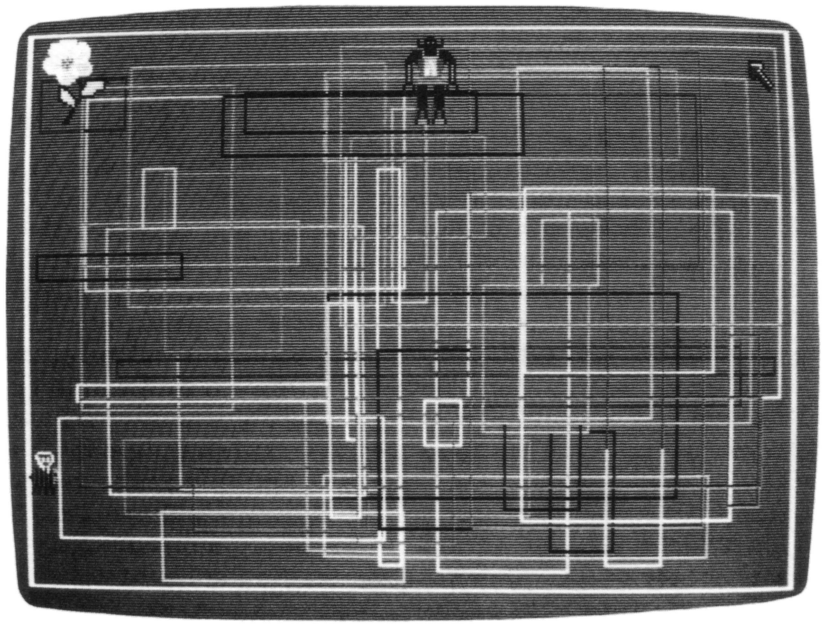


Figura 12-6.

Tre oggetti posizionati sullo schermo con i comandi OBJECT.X, OBJECT.Y e OBJECT.ON.

I COMANDI OBJECT.VX E OBJECT.VY

Per mettere in movimento gli oggetti bisogna usare i comandi OBJECT.VX e OBJECT.VY, che stabiliscono la velocità in senso orizzontale e verticale. La loro sintassi è:

```
OBJECT.VX numero di ID dell'oggetto, velocita' x

OBJECT.VY numero di ID dell'oggetto, velocita' y
```

Il numero di ID rappresenta un oggetto creato precedentemente con il comando OBJECT.SHAPE. Le velocità x e y possono essere indicate da un numero intero qualsiasi compreso fra -32768 e +32767.

Le due variabili indicano la velocità dell'oggetto in pixel al secondo: x è la velocità orizzontale e y quella verticale. Un valore positivo di x sposta l'oggetto verso destra, un valore negativo lo dirige verso sinistra. Un valore positivo di y muove l'oggetto verso il basso e, viceversa, un valore negativo lo sposta verso l'alto. Ad esempio,

```
OBJECT.VX 1, 23
OBJECT.VY 1, 42
```

indica una velocità che, in un secondo, muove l'oggetto 1 23 pixel a destra e 42 pixel più in basso rispetto alla posizione di partenza.

```
OBJECT.VX 1, -52
OBJECT.VY 1, -4
```

sposta l'oggetto 1 52 pixel a sinistra e 4 pixel più in alto, sempre nello spazio di un secondo.

Se volete essere molto precisi, potete applicare alcune regole fondamentali di trigonometria per calcolare la direzione in cui si muoverà l'oggetto dati i valori x e y. Ad esempio, considerate l'oggetto precedente con velocità x di 23 e velocità y di 42. Dividendo x per y e calcolando l'arcotangente del risultato, potete stabilire che l'oggetto si sposterà con una angolazione di 28,70 gradi, verso destra e verso il basso, nella direzione che ha la freccia nella Figura 12-7 (riportata nella pagina successiva).

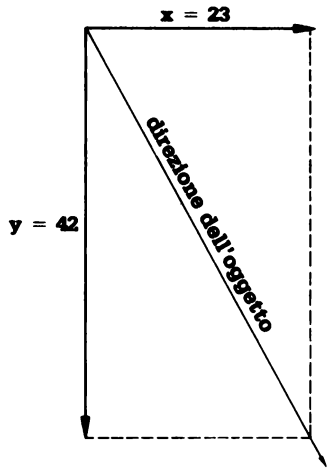


Figura 12-7.

La direzione di un oggetto che, in un secondo, si sposta di 23 pixel a destra e 42 pixel verso il basso.

Direzione dell'oggetto

Quando decidete la direzione di un oggetto con i valori x e y, ricordate che le proporzioni dei singoli pixel che formano lo schermo sono un fattore importante per stabilire esattamente la direzione voluta. Ad esempio, una velocità x di 5 e una medesima velocità y con uno schermo a risoluzione 1 fanno seguire al movimento dell'oggetto una diagonale di 45 gradi. La stessa velocità applicata a un oggetto inserito in uno schermo a risoluzione 2 determinerà però un movimento che segue un angolo di circa 30 gradi. La larghezza dei pixel è, in questo schermo, dimezzata rispetto all'altezza; ne risulta che i 5 pixel verticali costituiscono una distanza doppia rispetto ai 5 pixel orizzontali. Su uno schermo a risoluzione 3, i valori di x e y sposteranno l'oggetto secondo un angolo di 60 gradi, dato che i pixel in senso orizzontale misurano il doppio dei pixel in direzione verticale. La Figura 12-8 dimostra graficamente questo concetto.

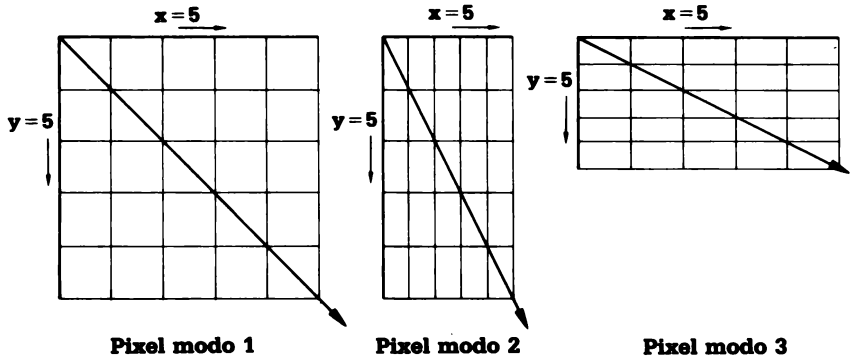


Figura 12-8.

I valori x e y della velocità producono risultati diversi a seconda del modo di risoluzione dello schermo.

Nel caso di un modo di risoluzione 4, l'angolo del movimento torna a essere di 45 gradi, come nello schermo a risoluzione 1, ma la velocità degli spostamenti risulterà dimezzata poiché, considerando la stessa distanza, il numero di pixel dello schermo ad alta risoluzione sarà doppio rispetto all'altro.

Anche se teoricamente è possibile fissare la velocità di un oggetto a 32767 pixel al secondo, di fatto non si possono utilizzare numeri così grandi. Con un'area di movimento di 320 per 200 pixel, qualsiasi valore di troppo superiore a 1000 pixel al secondo muove l'oggetto a una velocità tale da renderlo praticamente invisibile.

Per fissare la velocità degli oggetti del nostro esempio, aggiungete questa ulteriore sezione al programma:

```
OBJECT.VX 1, 10: OBJECT.VY 1, 10
OBJECT.VX 2, 0: OBJECT.VY 2, 10
OBJECT.VX 3, 60: OBJECT.VY 3, 0
```

Queste tre linee muovono l'oggetto 1 in direzione diagonale, in basso a destra, con un'angolazione di 45 gradi. Il secondo oggetto scorre invece verticalmente verso il basso, mentre il terzo corre velocemente verso destra.

IL COMANDO OBJECT.START

Fissare la velocità di un oggetto non significa ancora farlo muovere. A questo scopo bisogna usare il comando OBJECT.START con la sintassi seguente:

```
OBJECT.START numero di ID dell'oggetto, numero di ID dell'oggetto, ...
```

Il numero di ID indica uno degli oggetti realizzati dall'utente. OBJECT.START può essere seguito da tutti i numeri che una riga riesce a contenere, purché sempre separati da virgole. Il comando può anche essere usato con un solo numero o addirittura senza alcuna specificazione, nel qual caso vengono messi in movimento tutti gli oggetti presenti sullo schermo, alla velocità fissata con OBJECT.VX e OBJECT.VY. Specificando i numeri di ID si muoveranno invece solo gli oggetti indicati. Se l'utente non ha assegnato alcun valore x e y alla velocità di un oggetto, l'immissione del comando OBJECT.START non avrà alcun effetto sull'oggetto in questione, in quanto verrà dedotto che il valore di x e y è uguale a 0.

Mettendo in movimento un oggetto con OBJECT.START, esso si muove finché si scontra con un'altra figura o con i margini della finestra, o fino a quando viene fermato dal comando

OBJECT.STOP (di cui parleremo fra breve). Una volta fermo, l'oggetto resta immobile finché non viene riattivato da un altro comando OBJECT.START. Quando il programma finisce, si fermano tutti gli oggetti e, a meno che non inseriate un ciclo infinito o qualsiasi altro espediente per ritardare la fine del programma, tutti gli oggetti si fermeranno bruscamente non appena il BASIC ha eseguito tutte le linee di programma.

Per mettere in moto tutti gli oggetti del nostro esempio, aggiungete la linea sottostante alla fine della sezione *MoveObjects:*.

```
OBJECT.START
```

Se adesso volete vedere come funziona il programma, aggiungete la linea seguente alla fine del listato e fatelo girare:

```
Loop: GOTO Loop
```

In questo modo viene creato un ciclo infinito e gli oggetti continueranno a muoversi finché non incontreranno qualche ostacolo.

I COMANDI OBJECT.AX E OBJECT.AY

Dopo aver fissato la velocità di un oggetto e averlo messo in moto con OBJECT.START, esso si sposta nella direzione indicata a velocità costante. Per cambiare la direzione e la velocità l'utente può ricorrere ai comandi OBJECT.AX e OBJECT.AY. La loro sintassi è la seguente:

```
OBJECT.AX numero di ID dell'oggetto, accelerazione x
```

```
OBJECT.AY numero di ID dell'oggetto, accelerazione y
```

Il numero di ID si riferisce sempre a uno degli oggetti creati con il comando OBJECT.SHAPE. Le accelerazioni possono essere indicate da un numero intero qualsiasi compreso fra -32768 e +32767.

Le accelerazioni x e y indicano il cambiamento della velocità dell'oggetto misurata in pixel al secondo. Il valore x indica l'accelerazione orizzontale, mentre il valore y si riferisce all'accelerazione verticale. Un valore positivo di x accelera il movimento dell'oggetto verso destra e uno negativo verso sinistra. Un valore positivo di y accelera il movimento dell'oggetto verso il basso e uno negativo verso l'alto.

Per vedere come questi comandi di accelerazione influiscano sulla velocità, considerate un oggetto (numero di ID 2) con velocità x e y di 25 pixel al secondo. L'oggetto si muove con un'angolazione di 45 gradi verso l'angolo inferiore destro dello schermo. Introducete questi due comandi:

```
OBJECT.AX 2, -2
OBJECT.AY 2, -5
```

per modificare lentamente il movimento dell'oggetto, dirigendolo più verso l'alto e verso sinistra.

La tabella riportata qui sotto mostra i cambiamenti della velocità dell'oggetto in questione in un arco di 10 secondi.

	velocità x	velocità y
Velocità iniziale:	25	25
1 secondo:	23	20
2 secondi:	21	15
3 secondi:	19	10
4 secondi:	17	5
5 secondi:	15	0
6 secondi:	13	-5
7 secondi:	11	-10
8 secondi:	9	-15
9 secondi:	7	-20
10 secondi:	5	-25

Dopo 10 secondi, l'oggetto ha decisamente curvato per puntare verso l'alto con una leggera inclinazione a destra.

Applicando i cambiamenti di accelerazione alle direzioni fissate con i comandi OBJECT.VX e OBJECT.VY, è possibile far muovere gli oggetti secondo curve molto particolari. Fate un po' di prove per verificare le diverse possibilità offerte dall'accelerazione.

Per inserire qualche esempio interessante di accelerazione anche nel nostro programma, aggiungete la breve sezione che segue appena prima della sezione *Loop:* alla fine del listato:

```
ChangeMotion:
   OBJECT.AY 3, -2
```

Come risultato, l'oggetto numero 3 curva in alto verso la parte superiore dello schermo.

IL COMANDO OBJECT.STOP

Per fermare un oggetto in movimento bisogna utilizzare il comando OBJECT.STOP, che richiede la seguente sintassi:

```
OBJECT.STOP numero di ID dell'oggetto, numero di ID dell'oggetto, ...
```

Si possono specificare una quantità qualsiasi di oggetti, semplicemente elencando i loro numeri di ID e separandoli con virgole. Il comando OBJECT.STOP ferma solo gli oggetti specificati e, se non indicate alcun numero di ID, si bloccheranno tutte le figure in movimento sullo schermo.

Inserite la sezione seguente nel nostro programma dimostrativo, appena prima della sezione *Loop:*, e fate girare il programma per constatare che l'oggetto si ferma al centro del campo visivo:

```
StopObject:
    FOR i = 1 TO 3000: NEXT i
    OBJECT.STOP 2
```

Il ciclo FOR...NEXT interrompe momentaneamente il programma in modo che l'oggetto 2 possa muoversi parzialmente verso il basso prima di essere fermato dal comando OBJECT.STOP.

IL COMANDO OBJECT.OFF

Per rendere invisibile un oggetto che si trova nel campo visivo, l'utente deve utilizzare il comando OBJECT.OFF, il quale richiede la sintassi seguente:

```
OBJECT.OFF numero di ID dell'oggetto, numero di ID dell'oggetto, ...
```

Come in OBJECT.STOP, tutti gli oggetti possono essere specificati semplicemente indicando il loro numero di ID, e in questo caso il comando fa sparire solo gli oggetti specificati. Se non è seguito da nessun numero, il comando cancella tutti gli oggetti visualizzati in quel momento.

OBJECT.OFF blocca automaticamente il movimento degli oggetti prima di farli scomparire. Finché resta invisibile, un oggetto non può essere mosso, a meno che gli venga assegnata una nuova posizione con i comandi OBJECT.X e OBJECT.Y. Se non cambiate la locazione dell'oggetto prima di farlo riapparire utilizzando il comando OBJECT.ON, esso comparirà nello stesso punto in cui si era dileguato e da lì ricomincerà a muoversi.

Per verificare il funzionamento di OBJECT.OFF, sostituite la sezione *StopObject:* del nostro programma con le linee riportate nella pagina successiva, e poi fate girare il programma.

```
BlinkObjects:
    FOR i = 1 TO 3000: NEXT i
    OBJECT.OFF
    FOR i = 1 TO 3000: NEXT i
    OBJECT.ON
```

Il primo ciclo FOR...NEXT permette all'oggetto di muoversi per un certo tempo. Esaurito il ciclo, tutti gli oggetti dovrebbero sparire. A questo punto viene eseguito il secondo ciclo FOR...NEXT, al termine del quale le figure riappariranno nella stessa posizione in cui le avevate viste per l'ultima volta.

IL COMANDO OBJECT.CLOSE

Ogni volta che create un oggetto in memoria con il comando OBJECT.SHAPE, il BASIC utilizza parecchi kilobyte di memoria per memorizzarlo. Per disporre di questo spazio di memoria per altri scopi, potete cancellare gli oggetti che non vi servono più ricorrendo al comando OBJECT.CLOSE. La sua sintassi è:

OBJECT.CLOSE *numero di ID dell'oggetto, numero di ID dell'oggetto, ...*

OBJECT.CLOSE, come gli altri comandi, ha effetto solo sugli oggetti indicati dal relativo numero di ID. Se l'utente non specifica nessun numero, OBJECT.CLOSE cancella tutti gli oggetti in memoria.

Se viene specificato un oggetto che è ancora in movimento sullo schermo, esso si blocca e sparisce. Qualsiasi comando OBJECT successivo riferito al numero di ID di quell'oggetto non sortirà alcun effetto, a meno che non abbiate già creato, col comando OBJECT.SHAPE, un nuovo oggetto con lo stesso numero di ID. L'unico modo per riportare in memoria un oggetto cancellato consiste nel ricrearlo con il comando OBJECT.SHAPE.

Gli oggetti vengono conservati in memoria anche quando il programma di animazione ha smesso di girare; è quindi consigliabile cancellare tutti gli oggetti alla fine dell'esecuzione di un programma. Dal momento che molti programmi di animazione utilizzano un ciclo infinito per non interrompere il movimento delle figure, il programma deve spesso essere fermato con il comando Stop del menu Run; in questo caso è impossibile cancellare un oggetto attraverso il programma stesso. Dopo aver interrotto l'esecuzione del programma si potrebbe usare OBJECT.CLOSE nella finestra output, ma non è un metodo molto efficace per liberare la memoria. È meglio trovare un sistema alternativo per uscire dal programma, in modo da consentire a quest'ultimo di chiudere schermi e finestre, nonché di cancellare gli oggetti in memoria.

DUPLICAZIONE DI OGGETTI

Se scrivete un programma di animazione che si serve di una grande quantità di oggetti, è possibile realizzare duplicati degli oggetti già creati, risparmiando fatica e spazio di memoria. Il comando OBJECT.SHAPE può duplicare qualsiasi oggetto conservato in memoria; in seguito, il comando OBJECT.PLANES consente di modificare i colori dell'originale o delle riproduzioni.

COME USARE OBJECT.SHAPE PER LA DUPLICAZIONE DI OGGETTI

Quando usate OBJECT.SHAPE per creare un nuovo oggetto, assegnate la forma e il colore a un numero di ID, unitamente a una stringa di definizione che contiene i dati dell'oggetto. Se sostituite la stringa con il numero di ID di un oggetto già presente in memoria, OBJECT.SHAPE crea un nuovo oggetto con la stessa forma e lo stesso colore dell'originale. Per duplicare un oggetto, OBJECT.SHAPE utilizza la seguente sintassi:

```
OBJECT.SHAPE numero di ID del duplicato, numero di ID dell'originale
```

Il numero di ID dell'oggetto duplicato può essere un valore intero compreso fra 1 e il numero massimo di oggetti contenuti in memoria. Il numero di ID dell'originale deve indicare invece uno degli oggetti già creati con il comando OBJECT.SHAPE.

Lo spazio di memoria richiesto per un oggetto duplicato è inferiore allo spazio necessario a memorizzarne uno di nuova creazione. Ciò è dovuto al fatto che tanto l'originale quanto la sua copia utilizzano la stessa memoria per immagazzinare i dati sulla forma e sul colore. Ogni copia richiede solo lo spazio sufficiente per memorizzare la velocità e la locazione.

Tuttavia, ogni copia mossa sullo schermo costituisce un'unità a sé stante, con un numero di ID proprio; l'utente può assegnare valori di posizione, velocità e accelerazione diversi per ogni duplicato. Ogni copia può essere visualizzata e fatta sparire, messa in moto o fermata e, infine, cancellata, senza

che interferisca col comportamento delle altre copie o dell'originale. Però, se cancellate l'originale dalla memoria, i duplicati si bloccano e non rispondono a nessun tipo di comando. Fate dunque attenzione a non cancellare mai l'originale prima delle copie.

Il programma che segue crea un campo visivo e utilizza il comando OBJECT.SHAPE per fare tre copie di un oggetto. Colloca quindi i quattro oggetti negli angoli del campo visivo e li fa dirigere verso il centro:

```
PlayField:
    SCREEN 1, 320, 200, 5, 1
    WINDOW 2, , , 0, 1

MakeOriginal:
    OPEN ''Fiore.1bob3'' FOR INPUT AS 1
    OBJECT.SHAPE 1, INPUT$(LOF(1), 1)
    CLOSE 1

MakeDuplicates:
    FOR i = 2 TO 4
    OBJECT.SHAPE i, 1
    NEXT i

PlaceObjects:
    OBJECT.X 1, 0: OBJECT.Y 1, 0
    OBJECT.X 2, 0: OBJECT.Y 2, 150
    OBJECT.X 3, 250: OBJECT.Y 3, 150
    OBJECT.X 4, 250: OBJECT.Y 4, 0
    OBJECT.ON

MoveObjects:
    OBJECT.VX 1, 20: OBJECT.VY 1, 10
    OBJECT.VX 2, 20: OBJECT.VY 2, -10
    OBJECT.VX 3, -20: OBJECT.VY 3, -10
    OBJECT.VX 4, -20: OBJECT.VY 4, 10
    OBJECT.START

Loop: GOTO Loop
```

IL COMANDO OBJECT.PLANES PER CAMBIARE I COLORI

Le copie degli oggetti hanno la stessa forma e la stessa colorazione dell'originale. Per distinguere l'originale dal duplicato è possibile cambiare i colori dell'uno o dell'altro con il comando OBJECT.PLANES. OBJECT.PLANES funziona solo con i bob e non con gli sprite. La sua sintassi è:

OBJECT.PLANES *numero di ID dell'oggetto, valore di PlanePick, valore di PlaneOnOff*

Il numero di ID dell'oggetto può essere uno qualsiasi di quelli già usati con un comando OBJECT.SHAPE, indipendentemente dal fatto che si tratti di un originale o di una copia. I valori di PlanePick e PlaneOnOff devono essere indicati con numeri interi compresi fra 0 e 31 in uno schermo di 5 bit-plane, da 0 a 15 in uno di 4 bit-plane, da 0 a 7 in uno di 3 bit-plane, e da 0 a 3 in uno di 2 bit-plane. Il comando OBJECT.PLANES non ha alcun effetto su uno schermo con un solo bit-plane. È possibile omettere uno dei due valori (in questo caso esso viene automaticamente fissato a 0), ma non entrambi. Se si tralascia il valore di PlanePick, bisogna sostituirlo con una virgola.

OBJECT.PLANES utilizza le librerie di animazione grafica del software sistema per definire due maschere grazie ai valori PlanePick e PlaneOnOff indicati dall'operatore. Per capire come funzionano queste maschere di bit, dovete conoscere l'aritmetica binaria e il metodo di memorizzazione di dati grafici adottato dall'Amiga. Invece di addentrarsi in argomenti che vanno al di là delle finalità di questo libro, può essere d'aiuto descrivere in generale il funzionamento dei comandi OBJECT.PLANES e conoscere alcune regole empiriche che consentano un uso appropriato del comando, senza dover studiare argomenti troppo complessi. Se volete informazioni più dettagliate, consultate un manuale che tratti il ROM Kernel.

Per comprendere a grandi linee il meccanismo del comando OBJECT.PLANES, pensate al modo in cui l'Amiga memorizza i colori in un campo visivo. Il numero di registro del colore di ogni pixel è un numero binario contenuto in memoria con un bit in ogni bit-plane utilizzato dallo schermo. Ad esempio, in un campo visivo di 5 bit-plane per ogni pixel ha un numero di 5 bit, suddiviso in un bit per ciascun bit-plane. In un campo

visivo di 3 bit-plane, ogni pixel ha un numero di 3 bit, memorizzati uno in ogni bit-plane.

Ora pensate a cosa succede quando l'Amiga muove un bob di 2 bit-plane su uno schermo di 5 bit-plane. Per spostare il bob, l'Amiga cancella i pixel sui quali passa e li riempe con i valori del registro colore dei pixel del bob. Quando questo si sposta, l'Amiga reinserisce sul campo visivo i pixel originari. Dal momento che i pixel del bob sono profondi solo 2 bit-plane, i loro numeri di registro usano solo gli ultimi 2 bit-plane dei pixel che occupano sul campo visivo. I primi 3 bit-plane non vengono utilizzati, quindi l'Amiga li riempie con degli zeri.

Quando si fissa un valore PlanePick con OBJECT.PLANES, si ordina all'Amiga di distribuire i bit-plane del bob fra tutti i bit-plane del campo visivo. Ad esempio, con un bob di 2 bit-plane in un campo visivo di 5 bit-plane, potete fare in modo che i numeri di registro dei pixel del bob vengano memorizzati nei bit-plane 2 e 4 del campo visivo, invece che in 1 e 2 (i due bit-plane del fondo). L'Amiga assegnerà agli altri bit-plane (1,3 e 5) il valore di 0. Si ottiene così una nuova serie di numeri di registro del colore di ogni pixel del bob, che cambierà perciò colore.

È possibile cambiare ulteriormente i colori del bob fissando la maschera PlaneOnOff. Questa maschera riempie i bit-plane non utilizzati dai pixel del bob con 1 o 0, a seconda del valore di PlaneOnOff che avete utilizzato. Ciò vi permette di inserire 1 o 0 nei bit-plane non utilizzati dai pixel del bob e di cambiare nuovamente i suoi colori.

Un modo per ottenere risultati validi con il comando OBJECT.PLANES, senza calcolare i colori del bob originale e quelli del bob che ne deriva attraverso una conversione binaria dei bit-plane, consiste nel ricorrere a un metodo empirico. Scrivete un breve programma che visualizza sullo schermo un bob originale e una sua copia, e utilizzate OBJECT.PLANES con diversi valori di PlanePick e PlaneOnOff per cambiare i colori di entrambi. Fate girare il programma e osservate i risultati. Potete cambiare i valori di PlanePick e PlaneOnOff finché ottenete i risultati desiderati. Naturalmente, se state lavorando su uno schermo di 5 bit-plane, ci sono più di 1000 diverse combinazioni di valori, la maggior parte delle quali non darà i risultati sperati. Ad esempio, molti dei valori di PlanePick che potete utilizzare con OBJECT.PLANE diminuiranno in realtà il

numero di colori che desiderate per un bob. Ci sono diversi metodi per limitare il numero di valori possibili a una serie di valori che funzionino adeguatamente:

- Per evitare di perdere colori, non usate OBJECT.PLANE con un bob che abbia tanti bit-plane quanti lo schermo su cui è visualizzato. Altrimenti il numero dei colori del bob si ridurrà o resterà comunque invariato. Ad esempio, OBJECT.PLANE non è adatto per un bob di 4 bit-plane inserito in uno schermo con la stessa profondità.

- Il valore PlaneOnOff definisce una maschera che funziona indipendentemente dal valore della maschera PlanePick. Molti valori di PlaneOnOff non sortiscono alcun effetto quando lavorano con certi valori di PlanePick. È sempre meglio provare un valore di PlanePick separatamente per vedere come cambia il colore del bob, aggiungendo solo in seguito il valore di PlaneOnOff per vedere come altera i colori di PlanePick.

- Per riportare un bob alla sua colorazione originale, fissate i valori di PlanePick e di PlaneOnOff a 0.

Come stabilire i valori di PlanePick e PlaneOnOff

Quelle che seguono sono tabelle di valori PlanePick e PlaneOnOff che riportano valori corrispondenti a diverse profondità di bit-plane. Sono state preparate per farvi risparmiare tempo in tentativi vani e per non farvi commettere errori, dal momento che questi valori sono stati calcolati per ottenere buoni risultati. I valori di PlanePick non diminuiscono il numero dei colori del bob, mentre i valori di PlaneOnOff cambieranno i colori fissati dal valore di PlanePick.

Bob di 1 bit-plane

	Valori di PlanePick	Valori di PlaneOnOff che producono risultati
...in uno schermo di 2 bit-plane	1	0, 2
	2	0, 1
...in uno schermo di 3 bit-plane	1	0, 2, 4, 6
	2	0, 1, 4, 5
	4	0, 1, 2, 3
...in uno schermo di 4 bit-plane	1	0, 2, 4, 6, 8, 10, 12, 14
	2	0, 1, 4, 5, 8, 10, 12, 14
	4	0, 1, 2, 3, 8, 9, 10, 11
	8	0, 1, 2, 3, 4, 5, 6, 7
...in uno schermo di 5 bit-plane	1	0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30
	2	0, 1, 4, 5, 8, 9, 12, 13, 16, 17, 20, 21, 24, 25, 28, 29
	4	0, 1, 2, 3, 8, 9, 10, 11, 16, 17, 18, 19, 24, 25, 26, 27
	8	0, 1, 2, 3, 4, 5, 6, 7, 16, 17, 18, 19, 20, 21, 22, 23
	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15

Bob di 2 bit-plane

	Valori di Planepick	Valori di PlaneOnOff che producono risultati
...in uno schermo di 3 bit-plane	3	0, 4
	5	0, 2
	6	0, 1
...in uno schermo di 4 bit-plane	3	0, 4, 8, 12
	5	0, 2, 8, 10
	6	0, 1, 8, 9
	9	0, 2, 4, 6
	10	0, 1, 4, 5
	12	0, 1, 2, 3

...in uno	3	0, 4, 8, 12, 16, 20, 24, 28
schermo	5	0, 2, 8, 10, 16, 18, 24, 26
di 5	6	0, 1, 8, 9, 16, 17, 24, 25
bit-plane	9	0, 2, 4, 6, 18, 20, 22, 24
	10	0, 1, 4, 5, 16, 17, 20, 21
	12	0, 1, 2, 3, 16, 17, 18, 19
	17	0, 2, 4, 6, 8, 10, 12, 14
	18	0, 1, 4, 5, 8, 9, 12, 13,
	20	0, 1, 2, 3, 8, 9, 10, 11
	24	0, 1, 2, 3, 4, 5, 6, 7

Bob di 3 bit-plane

	Valori di Planepick	Valori di PlaneOnOff che producono risultati
...in uno	7	0, 8
schermo	11	0, 4
di 4	13	0, 2
bit-plane	14	0, 1
...in uno	7	0, 8, 16, 24
schermo	11	0, 4, 16, 20
di 5	13	0, 2, 16, 18
bit-plane	14	0, 1, 16, 17
	19	0, 4, 8, 12
	21	0, 2, 8, 10
	22	0, 1, 8, 9
	25	0, 2, 4, 6
	26	0, 1, 4, 5
	28	0, 1, 2, 3

Bob di 4 bit-plane

	Valori di PlanePick	Valori di PlaneOnOff che producono risultati
...in uno	15	0, 16
schermo	23	0, 8
di 5	27	0, 4
bit-plane	29	0, 2
	30	0, 1

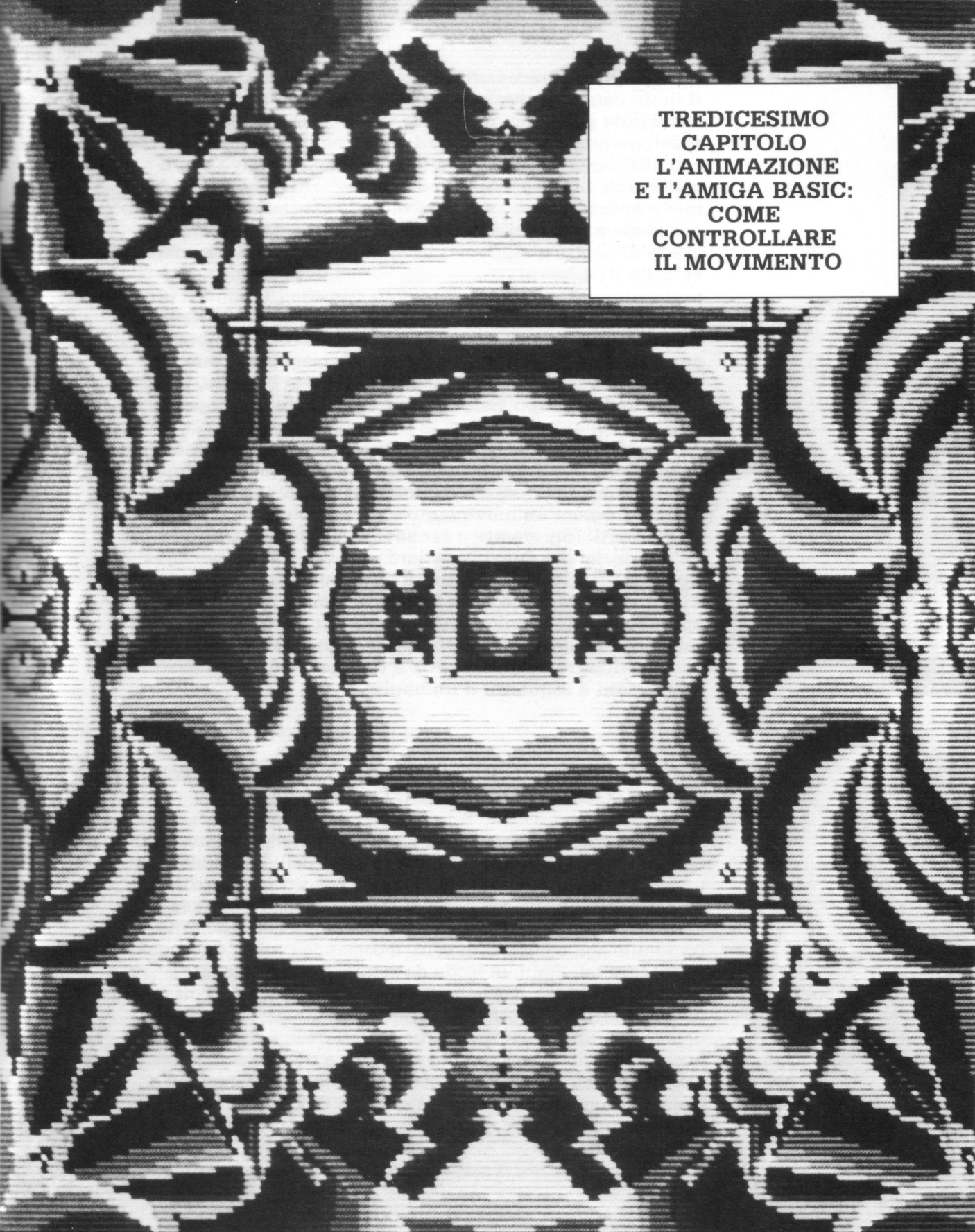
Un esempio di OBJECT.PLANES

Il miglior sistema per usare questi valori è trovare la tabella relativa al bob e allo schermo con cui state lavorando. Inserite i diversi valori nel comando OBJECT.PLANES e fate girare il programma per vedere i colori. Alla fine troverete i colori adatti alle vostre esigenze. La seguente sezione di programma utilizza tre serie di valori che cambiano il colore di tre bob di 3 bit-plane inseriti in uno schermo di 5 bit-plane.

```
ColorBobs:
    OBJECT.PLANES 2, 19, 8
    OBJECT.PLANES 3, 19, 0
    OBJECT.PLANES 4, 14, 1
```

Se inserite queste linee immediatamente prima della sezione *PlaceObjects:* del nostro programma dimostrativo, vedrete che ciascuna delle tre copie di bob viene colorata in maniera diversa.

Avete imparato anche a disegnare sprite e bob con l'Object Editor. Grazie ai comandi dell'Amiga BASIC illustrati in questo capitolo siete ora in grado di inserire i bob in memoria per poi visualizzarli sullo schermo, muovendoli in direzioni diverse e a varie velocità. Abbiamo visto come si procede alla duplicazione di un bob e alla modifica della sua colorazione. Il prossimo capitolo vi fornirà le informazioni necessarie a muovere i bob dove e come volete, controllando la loro reazione quando si imbattono in qualche ostacolo.



TREDICESIMO CAPITOLO L'ANIMAZIONE E L'AMIGA BASIC: COME CONTROLLARE IL MOVIMENTO

Nel capitolo precedente avete imparato a inserire oggetti in un campo visivo e a impartire loro le direttive necessarie per il movimento; li avete osservati durante lo spostamento, finché li avete fermati o si sono scontrati con altri oggetti o con i margini del campo visivo che li conteneva. L'Amiga BASIC può controllare il moto in modo ancor più preciso. Controllando costantemente la posizione e la velocità di un oggetto, l'Amiga BASIC è in grado di capire quando esso raggiunge la locazione o la velocità desiderata (o indesiderata), e di prendere provvedimenti adeguati. Controllando che gli oggetti non si scontrino fra loro o con i margini della finestra che li contiene, l'Amiga BASIC può fare in modo che due oggetti passino uno sull'altro senza alcuna conseguenza; può cambiare la direzione del movimento di un oggetto, può toglierlo dal campo visivo e compiere molte altre operazioni.

In questo capitolo verrà chiarito l'uso dei comandi di animazione dell'Amiga BASIC che controllano il movimento e le collisioni, nonché la coordinazione dei comandi al fine di ottenere un movimento armonioso degli oggetti visualizzati. Imparerete a far sovrapporre due oggetti in un'animazione a due dimensioni e mezzo e a evitare ogni tipo di contatto tra le figure in movimento sullo schermo. Alla fine del capitolo sono illustrate alcune interessanti applicazioni dell'Amiga BASIC: come far scorrere una sezione del campo visivo con un movimento di scroll e come utilizzare i comandi PUT e GET per realizzare vere animazioni a sequenza d'immagini.

DELIMITAZIONE DELL'AREA DI MOVIMENTO CON IL COMANDO OBJECT.CLIP

Uno dei modi più semplici per controllare il movimento degli oggetti in un campo visivo è rappresentato dalla creazione di un'area limitata di movimento. Quando create una finestra output per utilizzarla come campo visivo, il BASIC considera automaticamente il suo perimetro come limite del movimento consentito agli oggetti contenuti. Il comando OBJECT.CLIP può restringere tali limiti, in modo da evitare che l'oggetto si addentri in una sezione della finestra riservata a testi o disegni di contorno. OBJECT.CLIP può essere utilizzato anche per estendere la libertà di movimento oltre i limiti della finestra; in questo caso gli oggetti compiono un breve tragitto senza essere visti, da quando oltrepassano il confine della finestra output a quando incontrano il bordo fissato da OBJECT.CLIP.

IL COMANDO OBJECT.CLIP

Il comando OBJECT.CLIP ha la seguente sintassi:

OBJECT.CLIP *(indirizzo dell'angolo) - (indirizzo dell'angolo)*

Gli indirizzi degli angoli funzionano come gli indirizzi standard dei pixel usati con i comandi grafici, con la differenza che le coordinate x e y possono essere un numero intero qualsiasi compreso fra -32768 e +32767. È quindi possibile avere indirizzi che teoricamente sarebbero al di fuori della finestra output, come (-1045,78). Allo stesso modo di quelli utilizzati per creare un riquadro con un comando LINE, questi indirizzi indicano la posizione dei due angoli opposti di un quadrilatero.

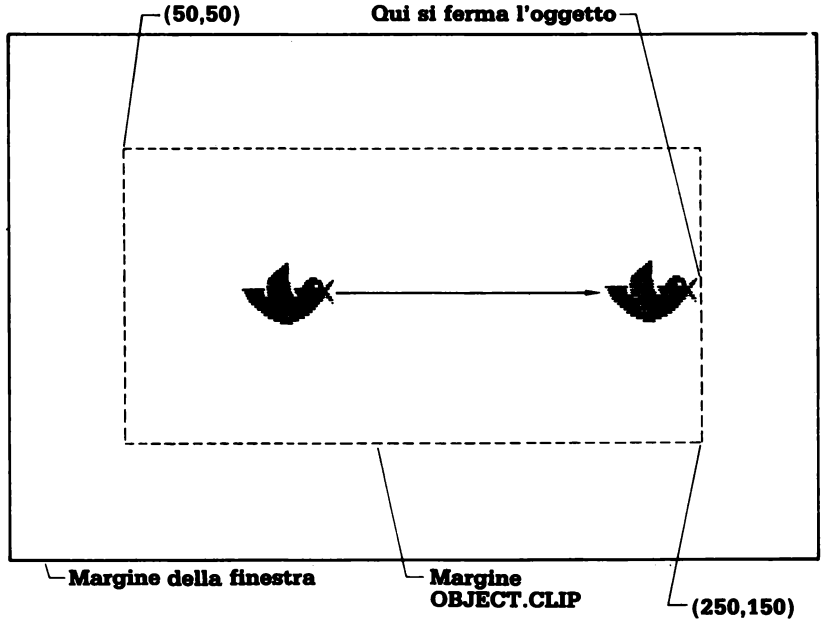
Qualsiasi oggetto che si muove all'interno del riquadro definito da OBJECT.CLIP si sposterà fino a incontrare uno dei margini oppure un altro oggetto. Gli oggetti posti invece al di fuori dell'area di movimento non si muovono assolutamente, neanche se viene assegnato loro un comando OBJECT.START.

Nell'esempio seguente, il comando OBJECT.CLIP crea un rettangolo invisibile al centro dello schermo. Quando fate girare il programma, vedrete un oggetto in movimento, che procederà nella sua corsa fino a essere fermato da uno dei margini. La Figura 13-1 (riportata nella pagina successiva) ne illustra il meccanismo.

```
MakePlayfield:
    SCREEN 1, 320, 200, 5, 1
    WINDOW 2, , , 0, 1

MakeObjects:
    OPEN ''Uccello.1bob1'' FOR INPUT AS 1
    OBJECT.SHAPE 1, INPUT$(LOF(1), 1)
    CLOSE 1

MoveObject:
    OBJECT.X 1, 100: OBJECT.Y 1, 100
    OBJECT.VX 1, 25
    OBJECT.CLIP(50,50) - (250,150)
    OBJECT.ON
    OBJECT.START

Loop: GOTO Loop
```

Figura 13-1.

Un oggetto viene bloccato dai margini fissati dal comando OBJECT.CLIP.



LETTURA DELLA POSIZIONE E DELLA VELOCITÀ DI UN OGGETTO

Per controllare gli oggetti più di quanto consenta di fare OBJECT.CLIP, è importante che il BASIC possa sempre leggere la posizione e la velocità di un oggetto in movimento. L'Amiga BASIC dispone di quattro funzioni che forniscono le coordinate x e y e la velocità x e y di ogni oggetto visualizzato nel campo visivo. Provate a scrivere un programma che verifichi i valori dati dalle funzioni e agisca di conseguenza.

LE FUNZIONI OBJECT.X() E OBJECT.Y()

OBJECT.X() e OBJECT.Y() sono due funzioni che forniscono le coordinate x e y della posizione di un oggetto.
La loro sintassi è:

```
OBJECT.X(numero di ID dell'oggetto)
OBJECT.Y(numero di ID dell'oggetto)
```

Fate attenzione a non confondere queste due funzioni con i comandi OBJECT.X e OBJECT.Y visti nel capitolo precedente. Le funzioni sono seguite dal numero di ID fra parentesi, senza specificazione di coordinate: è questo che le distingue dai comandi. Il numero di ID, come al solito, deve indicare uno degli oggetti creati con il comando OBJECT.SHAPE.

OBJECT.X() e OBJECT.Y() restituiscono le coordinate del pixel posto nell'angolo superiore sinistro dei margini dell'oggetto. La descrizione dei comandi OBJECT.X e OBJECT.Y e del modo in cui definiscono l'angolo superiore sinistro di un oggetto è riportata nel dodicesimo capitolo. OBJECT.X() restituisce la coordinata x (orizzontale), OBJECT.Y() la coordinata y (verticale).

Dal momento che queste due funzioni non sortiscono alcun effetto se usate da sole in una linea di programma, è necessario utilizzarle assegnando il loro valore a una variabile, oppure associandole a un comando. Ad esempio, le due linee seguenti assegnano le coordinate x e y dell'oggetto numero 3 alle variabili *x3* e *y3*:

```
x3 = OBJECT.X(3)
y3 = OBJECT.Y(3)
```

Se, nel momento in cui il BASIC esegue queste due funzioni, l'angolo superiore sinistro del margine dell'oggetto 3 si trova al pixel (45,102), il valore assegnato alla variabile *x3* sarà 45, quello della variabile *y3* corrisponderà a 102.

LE FUNZIONI OBJECT.VX() E OBJECT.VY()

È possibile leggere la velocità e la direzione degli oggetti utilizzando le funzioni OBJECT.VX() e OBJECT.VY() con la seguente sintassi:

```
OBJECT.VX(numero di ID dell'oggetto)
OBJECT.VY(numero di ID dell'oggetto)
```

Anche in questo caso, fate attenzione a non confondere queste funzioni con i comandi OBJECT.VX e OBJECT.VY visti nel capitolo precedente. Le funzioni indicano fra parentesi il numero di ID dell'oggetto, ma non accettano valori di velocità. Il *numero di ID* deve indicare un oggetto creato precedentemente con il comando OBJECT.SHAPE. OBJECT.VX() restituisce il valore della velocità orizzontale (x), OBJECT.VY() quello della velocità verticale (y), espresse naturalmente in pixel al secondo.

Come le due funzioni precedenti, anche OBJECT.VX() e OBJECT.VY() non possono essere usate da sole in una linea di programma. Devono necessariamente essere assegnate a una variabile o utilizzate assieme a un comando. Le due linee seguenti, ad esempio, leggono la velocità x e y dell'oggetto numero 2 e utilizzano i comandi OBJECT.VX e OBJECT.VY per assegnare i valori restituiti anche all'oggetto numero 3:

```
OBJECT.VX 3, OBJECT.VX(2)
OBJECT.VY 3, OBJECT.VY(2)
```

DIMOSTRAZIONE DELL'USO DELLE FUNZIONI DI LOCALIZZAZIONE E VELOCITÀ

Se create una serie di comandi in un ciclo che legge continuamente la posizione e la velocità degli oggetti, potete usare un comando IF...THEN per rimandare il programma a una certa sezione, quando i comandi inseriti nel ciclo leggono determinati valori della locazione o della velocità di un oggetto. Nell'esempio di programma riportato in basso, due oggetti si muovono nel campo visivo. L'oggetto 1 si sposta diagonalmente dall'angolo superiore sinistro verso l'angolo inferiore destro. L'oggetto 3 si sposta da sinistra a destra nella parte inferiore dello schermo, partendo lentamente e accelerando progressivamente.

La sezione di programma denominata *Test:* è un ciclo che utilizza OBJECT.Y() per controllare di quanto l'oggetto 1 si sposta verso la parte inferiore dello schermo, e OBJECT.VX per vedere a quale velocità si sta muovendo l'oggetto 3. Se la figura 1 (l'ape) giunge a una locazione la cui coordinata y è superiore a 99, il programma salta a una subroutine che ferma l'oggetto. Se la figura 3 (il fiore) raggiunge una velocità che supera i 50 pixel al secondo, il programma salta a una subroutine che impedisce al fiore di accelerare ulteriormente. La Figura 13-2 (nella pagina successiva) illustra il funzionamento di questa animazione.

```
MakePlayfield:
    SCREEN 1, 320, 200, 5, 1
    WINDOW 2, , , 0, 1

MakeObjects:
    OPEN ''Ape.spr'' FOR INPUT AS 1
    OBJECT.SHAPE 1, INPUT$(LOF(1), 1)
    CLOSE 1
    OPEN ''Fiore.1bob3'' FOR INPUT AS 1
    OBJECT.SHAPE 3, INPUT$(LOF(1), 1)
    CLOSE 1
```

(segue)

```
PlaceObjects:
    OBJECT.X 1, 0: OBJECT.Y 1, 0
    OBJECT.X 3, 0: OBJECT.Y 3, 150
    OBJECT.ON

MoveObjects:
    OBJECT.VX 1, 10: OBJECT.VY 1, 10
    OBJECT.VX 3, 5: OBJECT.VY 3, 1
    OBJECT.START

Test:
    IF OBJECT.Y(1) > 99 THEN GOSUB Stop1
    IF OBJECT.VX(3) > 60 THEN GOSUB Steady3
    GOTO Test

Stop1:
    OBJECT.STOP 1
    RETURN

Steady3:
    OBJECT.AX 3, 0
    RETURN
```

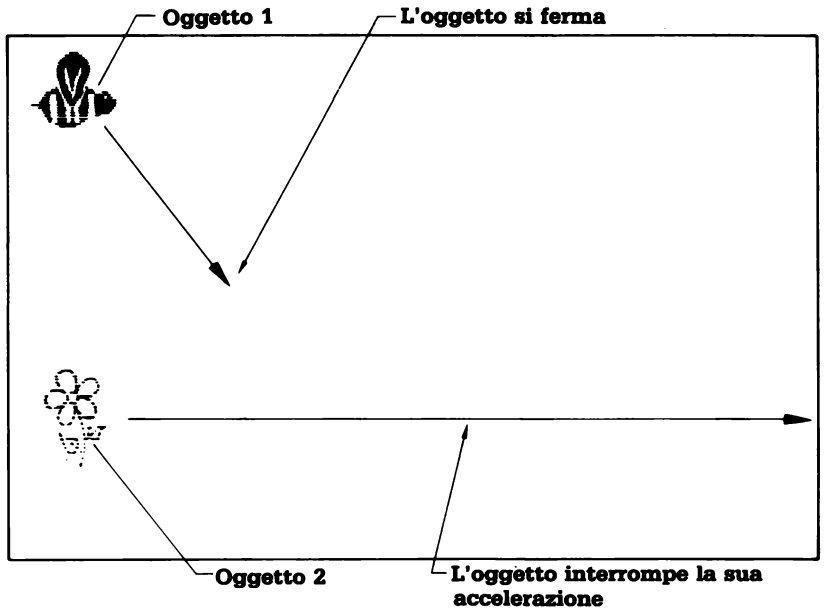


Figura 13-2.

Le funzioni di localizzazione e velocità del BASIC possono essere usate per controllare la posizione e la velocità di un oggetto, e per agire poi di conseguenza.

COME CONTROLLARE LE COLLISIONI

Nelle animazioni ottenute con l'Amiga BASIC è molto importante riuscire a gestire le collisioni che possono verificarsi. Come abbiamo visto nel capitolo precedente, quando un oggetto si scontra con un'altra figura o con i margini dell'area di movimento, si blocca automaticamente, a meno che non sia già stato introdotto un comando che indichi al BASIC quale azione compiere in caso di collisione. Il comando OBJECT.HIT consente di chiedere al BASIC di ignorare lo scontro, nel qual caso i due oggetti si oltrepassano senza nessuna conseguenza. Il comando OBJECT.PRIORITY dà la possibilità di decidere quale oggetto deve passare sopra all'altro. Se lasciate che gli oggetti si scontrino, potete usare i comandi COLLISION ON e ON COLLISION GOSUB, insieme alla funzione COLLISION(), per definire la reazione degli oggetti.

IL COMANDO OBJECT.HIT

Il comando OBJECT.HIT vi consente di definire diverse categorie di oggetti e di decidere quali categorie si scontreranno e quali no. Il comando utilizza la seguente sintassi:

```
OBJECT.HIT numero di ID dell'oggetto, valore della categoria, valore hit
```

Il numero di ID deve essere riferito a un oggetto creato con il comando OBJECT.SHAPE. Sia il valore della categoria che il valore hit possono essere indicati con numeri compresi fra -32768 e 32767.

Il primo valore definisce l'oggetto come appartenente a una sola, una qualsiasi o tutte le 16 categorie esistenti. L'Amiga BASIC interpreta la categoria 1 come il margine di una finestra o di un comando OBJECT.CLIP. Le altre 15 categorie sono astratte (il BASIC le distingue solo come numeri compresi fra 2 e 16), quindi potete farvene l'immagine che volete. Ad esempio, in un videogame potete decidere di usare gli oggetti della categoria 2 come astronavi, quelli della categoria 3 come raggi laser e quelli categoria 4 come pianeti. La tabella seguente indica il valore di ognuna delle 16 categorie disponibili:

Cat. 1: 1	Cat. 5: 16	Cat. 9: 256	Cat. 13: 4096
Cat. 2: 2	Cat. 6: 32	Cat. 10: 512	Cat. 14: 8192
Cat. 3: 4	Cat. 7: 64	Cat. 11: 1024	Cat. 15: 16384
Cat. 4: 8	Cat. 8: 128	Cat. 12: 2048	Cat. 16: -32768

Per definire un oggetto con una o più delle categorie, tranne la prima (non avrebbe senso definire un oggetto come margine e, inoltre, l'Amiga BASIC lo ignorerebbe), scegliete le categorie che volete, sommate i loro valori e otterrete il valore della categoria che vi interessa. Ad esempio, se volete che un oggetto venga definito sia come categoria 7 che come categoria 12, sommate i rispettivi valori (64 e 2048) e otterrete 2112, valore della categoria dell'oggetto in questione.

Se volete che un oggetto venga definito da una sola categoria, utilizzate solo il valore della categoria prescelta. Ad esempio, per far sì che un oggetto appartenga esclusivamente alla categoria 16, utilizzate il valore -32768. Specificare il valore 0 equivale a non assegnare all'oggetto nessuna delle 16 categorie, mentre il valore -1, somma di tutti i valori della tabella, definisce un oggetto come appartenente a tutte le categorie.

Il valore hit è invece un valore che identifica tutti i tipi di oggetto con cui la figura in questione si può scontrare. Quando un oggetto ne incontra un altro sul suo cammino, il BASIC controlla la categoria (o le categorie) del primo oggetto; quindi verifica il valore hit del secondo oggetto per vedere con quali altri oggetti esso può scontrarsi. Se il primo oggetto si trova nella lista di hit del secondo, il BASIC segnala una collisione. Se invece il primo oggetto non è compreso nella lista, le due figure si oltrepassano senza scontrarsi.

Il valore hit utilizza la stessa tabella di valori delle categorie. Per stabilire il valore hit, scegliete la categoria (o le categorie) di oggetti che volete inserire nella lista, sommate i loro valori e otterrete il valore di hit. Se non volete specificare nulla, utilizzate il valore 0; se volete invece inserire nella lista tutti gli oggetti, specificate il valore -1. Se volete considerare solo i margini, il valore da immettere è 1.

Quando due oggetti si toccano, è importante che il BASIC sappia qual è il primo e qual è il secondo, dal momento che deve confrontare il valore hit del primo con le categorie del secondo. La regola prevede che venga considerato primo oggetto quello che si trova più in alto a sinistra. Per fare in modo che un oggetto non si scontri con nessuno degli altri, dovete assegnargli un valore di categoria uguale a 0 e un

valore hit di 1, poiché non potete mai essere sicuri che esso venga controllato quale primo oggetto (verifica della categoria) o quale secondo (verifica della lista hit).

Quando un oggetto tocca un margine, l'Amiga BASIC controlla sempre la sua lista hit. Il margine non possiede tale lista, ha solo un valore di categoria uguale a 1.

Pensate ad esempio a un'animazione semplice, con tre tipi di oggetti che si muovono: api, fiori spazzati dal vento e uccelli. Le api non devono scontrarsi con altre api o con gli uccelli, bensì solo con i fiori, da cui devono succhiare il nettare. Gli uccelli non devono scontrarsi con nessuno dei tre tipi di oggetti e i fiori possono entrare in contatto solo con le api.

Se decidete che le api siano oggetti di categoria 2, gli uccelli di categoria 3 e i fiori di categoria 4, potete assegnare loro i rispettivi valori. Le api devono avere un valore di categoria uguale a 2 e un valore hit di 9, in modo da poter colpire sia i fiori (8) che i margini (1). Il valore di categoria degli uccelli dev'essere invece 4, mentre il valore hit deve corrispondere a 1, in modo che gli uccelli si scontrino solo con i margini. Ai fiori va assegnato un valore di 8 per la categoria e uno di 3 per la lista hit, in modo che possano toccare le api (2) e i margini (1).

Il programma dimostrativo che segue muove un'ape, un uccello e un fiore regolandosi con i valori di categoria e di hit. Vengono utilizzati gli sprite e i bob che avete disegnato nel dodicesimo capitolo. Quando fate girare il programma, vedrete l'ape (oggetto 1) muoversi diagonalmente sullo schermo, mentre l'uccello (oggetto 2) e il fiore (oggetto 3) si muovono in direzione orizzontale. L'ape passerà sopra l'uccello e si fermerà quando incontrerà il fiore. La Figura 13-3 (riportata nella pagina successiva) illustra le azioni degli oggetti.

```
MakePlayfield:
    SCREEN 1, 320, 200, 5, 1
    WINDOW 2, , , 0, 1

MakeObjects:
    OPEN ''Ape.spr'' FOR INPUT AS 1
    OBJECT.SHAPE 1, INPUT$(LOF(1), 1)
    CLOSE 1
    OPEN ''Uccello.1bob1'' FOR INPUT AS 1
    OBJECT.SHAPE 2, INPUT$(LOF(1), 1)
    CLOSE 1
    OPEN ''Fiore.1bob3'' FOR INPUT AS 1
    OBJECT.SHAPE 3, INPUT$(LOF(1) ,1)
    CLOSE 1
```

(segue)

```
PlaceObjects:
    OBJECT.X 1, 0: OBJECT.Y 1, 0
    OBJECT.X 2, 0: OBJECT.Y 2, 75
    OBJECT.X 3, 0: OBJECT.Y 3, 150
    OBJECT.ON

IdentifyObjects:
    OBJECT.HIT 1, 2, 9
    OBJECT.HIT 2, 4, 1
    OBJECT.HIT 3, 8, 3

MoveObjectsNew:
    OBJECT.VX 1, 20: OBJECT.VY 1, 20
    OBJECT.VX 2, 20
    OBJECT.VX 3, 20
    OBJECT.START

Loop: GOTO Loop
```

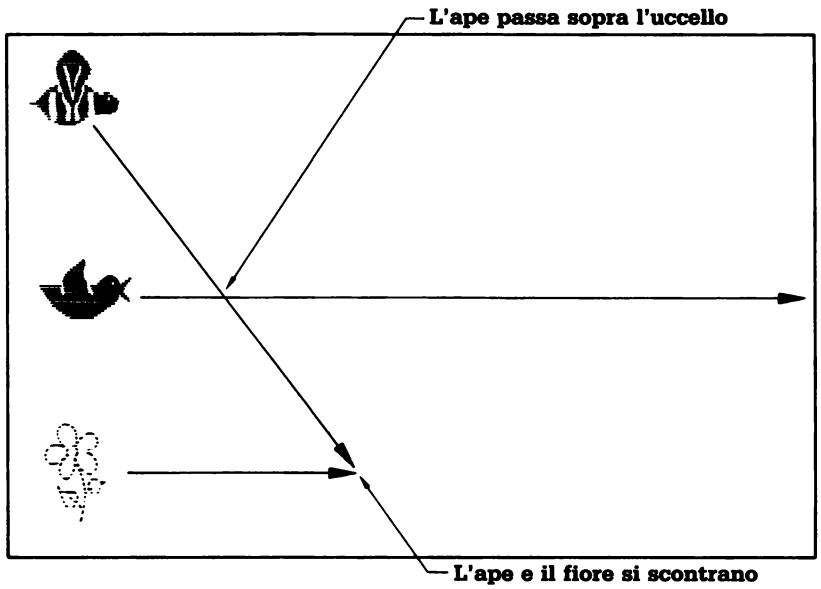


Figura 13-3.

Definendo diverse categorie di oggetti grazie al comando OBJECT.HIT, è possibile fare in modo che un'ape passi sopra a un uccello e vada a posarsi su un fiore.

IL COMANDO OBJECT.PRIORITY

Quando due oggetti entrano in contatto, il BASIC deve decidere quale dei due passerà al di sopra dell'altro. A questo scopo vengono utilizzate le stesse regole adottate per determinare quale fra due oggetti che si scontrano sarà il primo: l'oggetto più in alto a sinistra passerà perciò sopra a quello più

in basso a destra. Per modificare quest'ordine è necessario utilizzare il comando OBJECT.PRIORITY con la sintassi seguente:

```
OBJECT.PRIORITY numero di ID dell'oggetto, valore di precedenza
```

Il numero di ID indica uno degli oggetti creati dall'utente con il comando OBJECT.SHAPE. L'oggetto deve necessariamente essere un bob, poiché questo comando non funziona con gli sprite. Il valore di precedenza può essere un numero intero qualsiasi compreso fra -32768 e +32767.

Usando OBJECT.PRIORITY per assegnare un valore di precedenza a un bob, l'Amiga BASIC si ricorda di questo valore quando due oggetti si incontrano: quello con un valore di priorità più elevato passerà sopra l'altro. Ad esempio, se asegnate un valore di precedenza 2 a un bob e un valore di 2 a un altro, sarà quest'ultimo a passare sopra al primo. Se due bob hanno lo stesso valore torna valida la regola per cui ha la precedenza il bob più in alto a sinistra.

Pensate all'esempio precedente. Se sostituite nella sezione *MakeObjects:* lo sprite a forma di ape con il bob che riproduce il robot e aggiungete le linee che seguono subito prima di *MoveObjectsNew:*, il robot passerà sotto l'uccello perché il suo valore di priorità è inferiore.

```
SetPriority:
  OBJECT.PRIORITY 1, 0
  OBJECT.PRIORITY 2, 10
```

IL COMANDO COLLISION ON

Nonostante il nome possa indurre a pensarlo, il compito di COLLISION ON non è quello di rendere possibili gli scontri sullo schermo. Questo comando svolge la funzione di attivare la lista delle collisioni, cioè uno spazio di memoria che tiene il conto delle collisioni nell'ordine in cui avvengono.

COLLISION ON è inoltre la condizione basilare per il funzionamento di altri comandi e funzioni, come ON COLLISION GOSUB e COLLISION(), rispettivamente un comando e una funzione che si servono della lista delle collisioni. La sintassi del comando è assai semplice:

```
COLLISION ON
```

Una volta che l'Amiga BASIC esegue il comando, la lista viene attivata e resta tale finché non viene immesso il comando COLLISION OFF o il comando COLLISION STOP, di cui parleremo più avanti in questo capitolo.

IL COMANDO ON COLLISION GOSUB

Il BASIC ferma automaticamente un oggetto se questo si scontra con un'altra figura o con un margine (a meno che non abbiate dato istruzioni diverse). Se volete che, al momento della collisione, si verifichino altri effetti, potete scrivere una subroutine che realizzi l'azione desiderata e usare poi il comando ON COLLISION GOSUB per saltare a quella subroutine ogni volta che si verifica una collisione. Il comando richiede la seguente sintassi:

```
ON COLLISON GOSUB nome/numero della linea
```

Il nome o numero della linea deve indicare il nome della subroutine o il numero della sua linea d'inizio.

Una volta eseguito il comando ON COLLISION GOSUB, l'Amiga BASIC prosegue nell'esecuzione del resto del programma, come se non fosse successo niente, ma con un'eccezione. Ogni volta che esegue un comando, l'Amiga BASIC controlla la lista delle collisioni per vedere se ve ne sono state altre. Se l'esito è negativo, prosegue nell'esecuzione del comando successivo; se l'esito è positivo, il BASIC passa immediatamente a eseguire la subroutine indicata nel comando ON COLLISION GOSUB e, una volta completata quella, riprende dal comando successivo all'ultimo eseguito prima di registrare la collisione.

Se volete disattivare il comando ON COLLISION GOSUB in modo che, quando si verifica una collisione, il BASIC non salti alla subroutine, scrivete:

```
ON COLLISION GOSUB 0
```

Il numero di linea 0 interrompe il controllo delle collisioni.

Tutto ciò che avviene nella subroutine indicata nel comando dipende interamente dall'operatore. Ma prima di agire è meglio saperne di più sulla collisione che si è verificata: a tale scopo è possibile utilizzare la funzione COLLISION().

LA FUNZIONE COLLISION()

La funzione COLLISION() restituisce informazioni sulla lista delle collisioni, utilizzando la seguente sintassi:

`COLLISION`*(numero di ID)*

Il numero di ID deve indicare un oggetto creato precedentemente con un comando OBJECT.SHAPE; il numero può anche essere -1 o 0.

Per capire l'effetto della funzione COLLISION(), è importante sapere in quale modo la lista delle collisioni memorizza gli scontri. La lista può contenere fino a 16 collisioni alla volta. Se si verificano collisioni quando la lista è completa, esse non vengono memorizzate e non risulta che si siano mai verificate.

La lista memorizza tre informazioni per ogni collisione:

1. Il numero di ID del primo oggetto (quello più in alto a sinistra) fra i due che si scontrano. Se la collisione avviene fra un oggetto e un margine, viene memorizzato il numero di ID dell'oggetto.

2. Il numero di ID del secondo oggetto (quello più in basso a destra). Se la collisione avviene fra un oggetto e un margine, viene memorizzato un numero negativo che indica quale margine è stato toccato. La tabella seguente riporta i numeri e il rispettivo margine:

MARGINE	NUMERO DI ID
Superiore:	-1
Sinistro:	-2
Inferiore:	-3
Destro:	-4

3. Il numero della finestra in cui è avvenuta la collisione.

La lista memorizza le collisioni nell'ordine in cui avvengono. Quindi, la prima collisione della lista è anche la prima che si è verificata sullo schermo, l'ultima è quella più recente, come mostra la Figura 13-4.

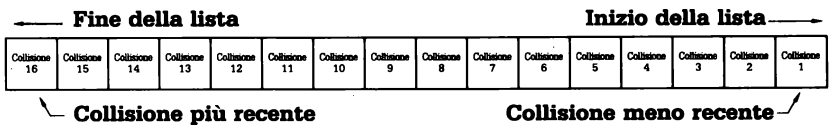


Figura 13-4.

La lista di collisione.

Se come numero di ID specificate uno 0, COLLISION() controlla la prima (più vecchia) collisione conservata nella lista e restituisce il numero di ID del primo oggetto coinvolto nello scontro. Se come numero di ID utilizzate invece -1, COLLISION() indica il numero della finestra in cui è avvenuto lo scontro. Quando controlla la lista utilizzando -1 o 0 come numero di ID, COLLISION() legge le informazioni dalla lista e non ne altera il contenuto.

Se inserite invece il numero di ID di un oggetto (numero intero positivo), COLLISION() cerca all'interno della lista una collisione in cui l'oggetto in questione era coinvolto, indipendentemente dal fatto che fosse considerato primo o secondo oggetto. Se la trova, COLLISION() restituisce il numero di ID dell'oggetto o del margine con cui l'oggetto in questione si è scontrato. Inoltre, la funzione elimina la collisione appena letta dalla lista, facendo scorrere tutte le altre di una posizione e creando un posto libero alla fine della lista. Se COLLISION() non trova alcuna collisione che coinvolga l'oggetto specificato, restituisce uno 0 e non altera il contenuto della lista.

L'esempio seguente vi dà un'idea del funzionamento della funzione COLLISION(). Su di un campo visivo si verificano tre collisioni in sequenza, con la lista delle collisioni attivata. L'oggetto 2 si scontra con il numero 1, mentre il 4 urta il margine sinistro e il 3 si imbatte nell'oggetto numero 1. Se scrivete

```
PRINT COLLISION(0)
```

per controllare il contenuto della lista, verrà indicato un 2, il numero di ID del primo oggetto coinvolto nella prima collisione. Se poi immettete

```
PRINT COLLISION(2)
```

il valore restituito sarà 1, numero di ID dell'oggetto colpito dall'oggetto 2, e la registrazione di tale collisione verrà eliminata dalla lista. Immettendo

```
PRINT COLLISION(1)
```

COLLISION() cerca all'interno della lista uno scontro che abbia coinvolto l'oggetto 1, trovandolo alla fine della lista; la funzione restituisce così il valore 3, numero di ID dell'oggetto colpito dal numero 1, ed elimina la collisione in questione dalla lista.

L'unica collisione ancora memorizzata (passata ora al primo posto della lista) è quella fra l'oggetto 4 e il margine sinistro. Se scrivete

```
COLLISION(4)
```

viene restituito un valore di -2, numero di ID del margine coinvolto, e anche quest'ultima collisione viene cancellata dalla lista.

DIMOSTRAZIONE DELL'USO DI COLLISION ON, ON COLLISION GOSUB E COLLISION()

L'esempio di programma riportato più avanti posiziona tre oggetti sullo schermo e li muove in diverse direzioni. La sezione denominata *DetectCollision:* utilizza COLLISION ON per attivare la lista delle collisioni e ON COLLISION GOSUB per fare in modo che il programma inizi a controllare gli scontri. In caso di collisione viene ordinato di saltare alla linea denominata *CheckCollision:*.

La subroutine *CheckCollision:* controlla i tipi di collisioni avvenute e modifica la direzione degli oggetti in modo che si allontanino da tutto ciò che colpiscono. La seconda linea della routine trova il numero del primo oggetto coinvolto nel primo scontro e lo memorizza nella variabile *objectid*. La terza linea controlla che ci sia stata effettivamente una collisione e, in caso contrario, riporta l'esecuzione del programma alla sezione *Loop:*, un ciclo infinito che continua a girare finché non si verifica uno scontro. La quarta linea di *CheckCollision:* legge il numero di ID dell'oggetto colpito nella prima collisione e lo assegna alla variabile *hitobject*.

Le due linee successive controllano se l'oggetto ha colpito il margine superiore o inferiore (-1 o -3) e, se la risposta è positiva, invertono il movimento verticale dell'oggetto in causa. Le righe 7 e 8 controllano analogamente se l'oggetto ha colpito il margine sinistro o destro (-2 o -4), invertendo eventualmente il movimento orizzontale. Le linee 9, 10 e 11 deducono che, se l'oggetto non ha toccato alcun margine, deve necessariamente aver toccato un altro oggetto, e quindi cambiano il movimento verticale e orizzontale di entrambi gli oggetti coinvolti nello scontro.

Il comando OBJECT.START, inserito verso la fine della subroutine, rimette in movimento l'oggetto o gli oggetti coinvolti nella collisione e bloccati automaticamente dal BASIC non appena si sono scontrati. Infine l'ultima linea del programma che segue ne riporta l'esecuzione alla sezione *Loop:*.

```
MakePlayfield:
    SCREEN 1, 320, 200, 5, 1
    WINDOW 2, , , 0, 1

MakeObjects:
    OPEN ''Ape.spr'' FOR INPUT AS 1
    OBJECT.SHAPE 1, INPUT$(LOF(1), 1)
    CLOSE 1
    OPEN ''Uccello.1bob1'' FOR INPUT AS 1
    OBJECT.SHAPE 2, INPUT$(LOF(1), 1)
    CLOSE 1
    OPEN ''Fiore.1bob3'' FOR INPUT AS 1
    OBJECT.SHAPE 3, INPUT$(LOF(1), 1)
    CLOSE 1

PlaceObjects:
    OBJECT.X 1, 0: OBJECT.Y 1, 0
    OBJECT.X 2, 0: OBJECT.Y 2, 150
    OBJECT.X 3, 250: OBJECT.Y 3, 0
    OBJECT.ON

MoveObjects:
    OBJECT.VX 1, 15: OBJECT.VY 1, 25
    OBJECT.VX 2, 40: OBJECT.VY 2, -30
    OBJECT.VX 3, -15: OBJECT.VY 3, 20
    OBJECT.START

DetectCollision:
    COLLISION ON
    ON COLLISION GOSUB CheckCollision

Loop: GOTO Loop

CheckCollision:
    objectid = COLLISION(0)
    IF objectid = 0 THEN RETURN
    hitobject = COLLISION(objectid)
    IF hitobject = -1 OR hitobject = -3 THEN
        OBJECT.VY objectid, -(OBJECT.VY(objectid))
    ELSEIF hitobject = -2 OR hitobject = -4 THEN
        OBJECT.VX objectid, -(OBJECT.VX(objectid))
    ELSE
        OBJECT.VY objectid, -(OBJECT.VY(objectid))
        OBJECT.VX objectid, -(OBJECT.VX(objectid))
        OBJECT.VY hitobject, -(OBJECT.VY(hitobject))
        OBJECT.VX hitobject, -(OBJECT.VX(hitobject))
    END IF
    OBJECT.START
    RETURN
```

IL COMANDO COLLISION STOP

A volte può essere necessario evitare che ON COLLISION GOSUB rimandi alla subroutine a ogni collisione. Per sospendere temporaneamente ON COLLISION GOSUB potete usare il comando COLLISION STOP, rispettando la seguente sintassi:

```
COLLISION STOP
```

Quando il BASIC esegue COLLISION STOP dopo un comando ON COLLISION GOSUB, la lista delle collisioni viene costantemente aggiornata finché è completa. Però non viene comunicato a ON COLLISION GOSUB che si sono verificati degli scontri, in modo da evitare che il programma salti alla subroutine.

Per rimettere in funzione ON COLLISION GOSUB dovrete utilizzare un altro comando COLLISION ON. Se nel frattempo la lista si è arricchita di nuove collisioni, il comando ON COLLISION GOSUB appena riattivato le leggerà e passerà all'esecuzione della subroutine.

IL COMANDO COLLISION OFF

Il comando COLLISION OFF disattiva completamente la lista delle collisioni, con questa semplice sintassi:

```
COLLISION OFF
```

Una volta eseguito questo comando, le collisioni non vengono più memorizzate nella lista e l'Amiga BASIC le dimentica completamente. Nella lista rimangono comunque tutte le collisioni immagazzinate prima dell'esecuzione di COLLISION OFF. Per riprendere la memorizzazione non dovete fare altro che immettere un nuovo comando COLLISION ON.

ALTRI TRUCCHI PER L'ANIMAZIONE SULL'AMIGA

La maggior parte dei comandi di animazione dell'Amiga BASIC sono comandi OBJECT che muovono sprite e bob sullo schermo. L'Amiga BASIC presenta altre due possibilità per simulare il movimento: lo scorrimento dello sfondo e l'animazione a sequenza d'immagini.

SCORRIMENTO DEL CAMPO VISIVO E COMANDO SCROLL

Il comando SCROLL consente di definire un'area all'interno del campo visivo e di far scorrere il suo contenuto verticalmente, orizzontalmente o diagonalmente. La sintassi è la seguente:

SCROLL *(indirizzo dell'angolo) - (indirizzo dell'angolo), spostamento x, spostamento y*

Gli indirizzi degli angoli prevedono le stesse modalità del comando OBJECT.CLIP. Le coordinate x e y possono essere rappresentate da un numero qualsiasi compreso fra -32768 e +32767, rendendo perciò possibili indirizzi tipo (-1045,78). Questi due indirizzi stabiliscono la posizione di due angoli opposti di un rettangolo che può essere fatto scorrere. Gli spostamenti x e y possono essere indicati da qualsiasi numero intero incluso fra -32768 e +32767. È possibile utilizzare solo uno spostamento x o uno y; se tralasciate il valore di x, sostituitelo con una virgola.

I due indirizzi fissano gli angoli opposti di un'area rettangolare all'interno della finestra output. L'area creata col comando SCROLL può essere più piccola della finestra output o può occuparla interamente. Anche se specificate indirizzi al di fuori della finestra, saranno comunque considerati quali margini del rettangolo i bordi della finestra stessa. Ad esempio, se immettete un comando SCROLL con indirizzi (-256,-45) - (600,450) in una finestra a risoluzione 1, non fate altro che creare un rettangolo grande quanto la finestra, non un pixel di più.

Il valore dello spostamento x indica la distanza in pixel lungo la quale volete muovere il contenuto del rettangolo, naturalmente in senso orizzontale. Un numero positivo causa uno scroll verso destra, uno negativo verso sinistra. Il valore dello spostamento y indica la distanza in pixel lungo la quale volete spostare il contenuto del rettangolo, questa volta in senso verticale. Un numero positivo determina un movimento verso il basso, uno negativo verso l'alto. Utilizzando contemporaneamente i due valori x e y, si ottiene uno scorrimento diagonale. Ad esempio, il seguente comando sposta il contenuto di un rettangolo situato nell'angolo superiore sinistro della finestra output, muovendolo di 20 pixel verso destra e 30 pixel verso il basso.

```
SCROLL (0,0) - (160,100), 20, 30
```

Quando l'Amiga BASIC esegue questo comando, il contenuto dell'area indicata si sposta diagonalmente verso l'angolo inferiore destro della finestra.

Spostando il contenuto dell'area al di fuori dei margini della finestra non si fa altro che cancellarlo. Se fate scorrere il rettangolo dell'esempio precedente nella posizione opposta, la parte che oltrepassa il margine risulta completamente vuota. Ad esempio,

```
SCROLL (0,0) - (160,100), 20, 30
SCROLL (0,0) - (160,100), -20, -30
```

fa scorrere il contenuto del rettangolo oltre i suoi margini e poi lo riporta nella sua posizione originale. La Figura 13-5 (qui di seguito) mostra i risultati.

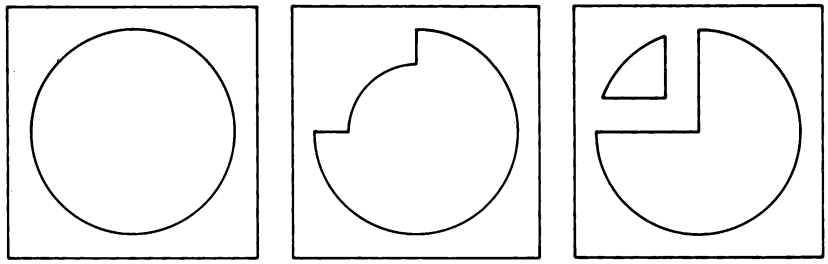


Figura 13-5.

Tre diverse fasi di uno scorrimento. L'immagine di sinistra mostra la situazione di partenza, la figura centrale riproduce la finestra dopo l'esecuzione del primo comando SCROLL, l'immagine di destra mostra l'aspetto definitivo della finestra dopo l'ultimo comando SCROLL.

Un unico comando SCROLL non fa scorrere il contenuto del rettangolo; lo sposta semplicemente con un movimento istantaneo. Per ottenere un chiaro effetto di scorrimento, dovete usare più comandi SCROLL in successione, con valori di spostamento x e y molto bassi. Ad esempio, il programma che segue crea uno sfondo e poi ne fa scorrere una sezione avanti e indietro in direzione diagonale, utilizzando una serie di cicli FOR...NEXT:

```
MakePlayfield:
    SCREEN 1, 320, 200, 5, 1
    WINDOW 2, , , 0, 1

FillText:
    WIDTH 40
    FOR i = 1 TO 100
        PRINT ''Kumquat '';
    NEXT i

ScrollBlock:
    FOR i = 1 TO 15
        FOR j = 1 TO 20: NEXT j
        SCROLL (100, 25) - (200,110), -1, -1
    NEXT i
    For i = 1 TO 15
        For j = 1 TO 20 : NEXT j
```

(segue)

```
    SCROLL (100, 25) - (200, 110), -1, -1
NEXT i
GOTO ScrollBlock
```

Il primo ciclo FOR...NEXT della sezione *ScrollBlock:* fa scorrere il contenuto del rettangolo verso destra e verso il basso. L'altro breve ciclo FOR...NEXT fa in modo che lo scorrimento non sia troppo veloce. Per accelerarlo è sufficiente sostituire 20 con un valore più basso, per rallentarlo bisogna inserirne uno più alto. Il successivo ciclo FOR...NEXT riporta il contenuto nella sua posizione originaria, regolando ancora una volta la velocità di scorrimento con un secondo ciclo.

Il comando SCROLL può essere utilizzato per muovere il contenuto del campo visivo anche quando questo è occupato da oggetti. SCROLL, infatti, farà scorrere lo sfondo senza disturbare le figure di primo piano. Se gli oggetti visualizzati sono bob, potreste però avere effetti strani. Dato che i bob vengono effettivamente disegnati sul campo visivo, è possibile che lascino uno strano "residuo grafico" quando il campo scorre sotto di loro. Gli sprite non costituiscono invece un problema perché non fanno assolutamente parte dello sfondo.

IL COMANDO PUT PER L'ANIMAZIONE SEQUENZIALE

Sprite e bob sono mezzi ottimi per il movimento esterno, grazie anche ai comandi dell'Amiga BASIC che facilitano il controllo del moto, ma non hanno il minimo movimento interno. È possibile affiancare più sprite o più bob per realizzare un oggetto di grandi dimensioni e dotato di movimento interno, ma può risultare alquanto difficile allineare e muovere i singoli componenti; bisogna infatti mantenere costantemente sotto controllo la posizione e il movimento di moltissimi oggetti separati. Le tecniche di animazione messe a disposizione dal comando PUT rendono tutto più semplice.

Nel sesto capitolo abbiamo visto come GET e PUT possano memorizzare un blocco grafico in un array per reinserirlo in un punto diverso della finestra output. Abbiamo anche visto come bisogna utilizzare GET per immagazzinare una serie di blocchi grafici in un array bidimensionale. Se memorizzate una serie di blocchi che contengono una serie di immagini in sequenza, potrete poi ricorrere a PUT per visualizzare tutti i blocchi in rapida successione, creando così oggetti dotati di movimento esterno e interno.

Il programma della pagina seguente utilizza GET per memorizzare quattro fasi del volo di un uccello. In un secondo tempo il comando PUT, inserito in due cicli FOR...NEXT, esegue una rotazione delle immagini dando l'impressione che l'uccello voli. La Figura 13-6 (riportata a pagina 373) mostra il movimento realizzato da questo programma.

```
MakeWindow:
    SCREEN 1, 320, 200, 2, 1
    WINDOW 2, , , 0, 1
CreateArray:
    DIM bird(1440,3)
DrawFrames:
    frame = 0
    LINE (160,100) - STEP(-40,-40), 3
    LINE (160,100) - STEP(40,-40), 3
    GOSUB GetFrame
    CLS

    frame = 1
    LINE (160,100) - STEP(-50,-15), 3
    LINE (160,100) - STEP(50,-15), 3
    GOSUB GetFrame
    CLS

    frame = 2
    LINE (160,100) - STEP(-50,15), 3
    LINE (160,100) - STEP(50,15), 3
    GOSUB GetFrame
    CLS

    frame = 3
    LINE (160,100) - STEP(-40,40), 3
    LINE (160,100) - STEP(40,40), 3
    GOSUB GetFrame
    CLS

    x = 0: y = 0
SequenceFrames:
    FOR i = 0 TO 3
        PUT (x,y), bird(0,i)
        FOR j = 1 TO 200: NEXT j
        PUT (x,y), bird(0,i)
    NEXT i
    FOR i = 2 TO 1 STEP -1
        PUT (x,y), bird(0,i)
        FOR j = 1 TO 200: NEXT j
        PUT (x,y), bird(0,i)
    NEXT i
    x = x + 5: y = y + 5
    GOTO SequenceFrames
GetFrame:
    GET (100,55) - (220,145), bird(0,frame)
    RETURN
```

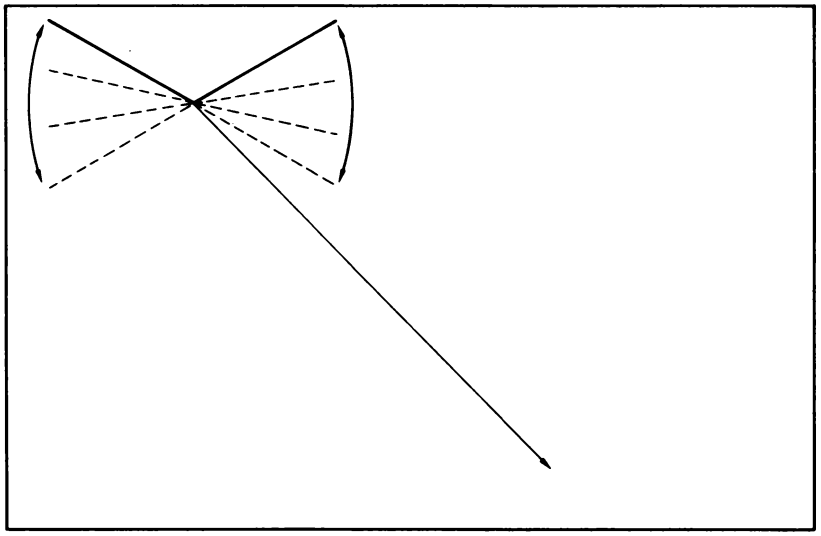


Figura 13-6.

L'uccello realizzato dal programma dimostrativo.

Quando fate girare il programma, vedrete un breve flash al centro della finestra mentre le quattro fasi della sequenza di volo vengono disegnate e salvate con GET. L'uccello stilizzato (molto essenziale) apparirà nell'angolo superiore sinistro dello schermo e si sposterà verso l'angolo inferiore destro sbattendo le ali, fino a superare i margini della finestra output.

Per capire come funziona il programma, osservate le sue parti costituenti. *MakeWindow:* crea uno schermo a bassa risoluzione con 2 bit-plane e una finestra a tutto schermo. *CreateArray:* definisce un array bidimensionale per contenere l'uccello e gli assegna il nome *bird*. La sezione seguente, *DrawFrames:*, è divisa in quattro parti, una per ogni fotogramma del volo. Ogni parte stabilisce un numero da assegnare a ciascun fotogramma e poi disegna il volatile utilizzando due semplici comandi LINE. Disegnato l'uccello, richiama la subroutine *GetFrame:* che salva il blocco contenente l'uccello nell'array creato precedentemente. Poi lo schermo viene liberato per la fase successiva.

La sezione successiva, *SequenceFrames:*, esegue la rotazione dei fotogrammi per realizzare l'animazione. Per posizionare l'uccello, la linea che precede la sezione stabilisce le coordinate x e y che posizionano il blocco PUT.

I due cicli FOR...NEXT compresi nella sezione *SequenceFrame:* fanno sbattere le ali: il primo ciclo verso il basso e il secondo verso l'alto. In ogni ciclo ci sono due comandi PUT identici, separati da un breve ciclo FOR...NEXT. Il primo comando PUT disegna l'uccello sullo schermo; il ciclo FOR...NEXT mantiene l'immagine sullo schermo; il secondo comando PUT disegna la stessa figura nella stessa posizione, con l'effetto di

cancellare l'uccello dallo schermo (ricordate che PUT utilizza l'opzione XOR per disegnare la figura, col risultato che una sovrapposizione di due immagini identiche non fa che cancellare la prima immagine visualizzata, come già spiegato dettagliatamente nel sesto capitolo).

Alla fine dei due cicli è stata inserita una linea che aumenta il valore delle coordinate x e y di 5, cambiando così la posizione del blocco PUT a ogni passaggio del ciclo; in questo modo l'uccello si muove a destra e verso il basso a ogni colpo d'ala. La linea successiva riporta il programma a *SequenceFrames:*, che riprende il battito d'ali. Questo tipo di animazione presenta aspetti positivi e negativi. Lo spazio di memoria richiesto è piuttosto consistente, specialmente se usate uno schermo con molti bit-plane, una figura grande e parecchi fotogrammi. Quando disegnate le figure originali si nota una certa vibrazione dello schermo, e il lavoro richiesto per disegnare figure più realistiche del nostro uccello stilizzato non è certo da sottovalutare. Questa tecnica di animazione consente però di completare un oggetto col movimento interno. Per eliminare il tremolio nella fase di disegno, potete realizzare le immagini in un programma a parte, salvarle e poi caricarle nel programma di animazione. Naturalmente serve un po' di pratica con i comandi Amiga BASIC per la gestione dei file.

In quest'ultimo capitolo avete visto com'è possibile limitare i movimenti di un oggetto entro una certa area e leggerne la velocità e la posizione. Avete anche imparato a gestire le collisioni fra gli oggetti, a far scorrere il campo visivo e a utilizzare l'animazione sequenziale. Sarà necessaria un po' di pratica per assimilare tutto ciò che avete appreso sui comandi di animazione, ma quando saprete usarli in modo creativo e originale, le vostre animazioni computerizzate desteranno grande interesse e stupore.

CONCLUSIONE

IL FUTURO: LE POSSIBILITÀ CREATIVE DELL'AMIGA

Dopo aver imparato a usare con una certa dimestichezza Deluxe Paint, Deluxe Music, Deluxe Video e gli altri strumenti grafici, musicali e di animazione dell'Amiga avete paura di trovarvi davanti allo spettro della noia e di non avere assolutamente più nulla di nuovo da imparare? È un'ipotesi alquanto improbabile. I programmi attualmente in vendita per l'Amiga sono solo l'inizio. Il numero di programmatori che impara a sfruttare al meglio le capacità dell'Amiga aumenta ogni giorno, e aumenta quindi anche il software offerto agli utenti.

L'Amiga ha un grande potenziale non ancora sfruttato a pieno. Ad esempio, una delle caratteristiche meno sfruttate dell'Amiga è il modo HAM (Hold and Modify) per la visualizzazione grafica, che consente al computer di utilizzare contemporaneamente tutti i 4096 colori di cui dispone (con poche eccezioni). Nonostante alcuni digitalizzatori realizzino immagini HAM contraddistinte da sfumature e effetti a tutto tondo davvero stupefacenti, esistono ancora pochi programmi grafici che consentano di realizzare immagini HAM partendo da zero e di riprodurle su carta. Ma con il costante aumento della domanda di programmi che utilizzino molti colori, le case produttrici non tarderanno a soddisfare le richieste.

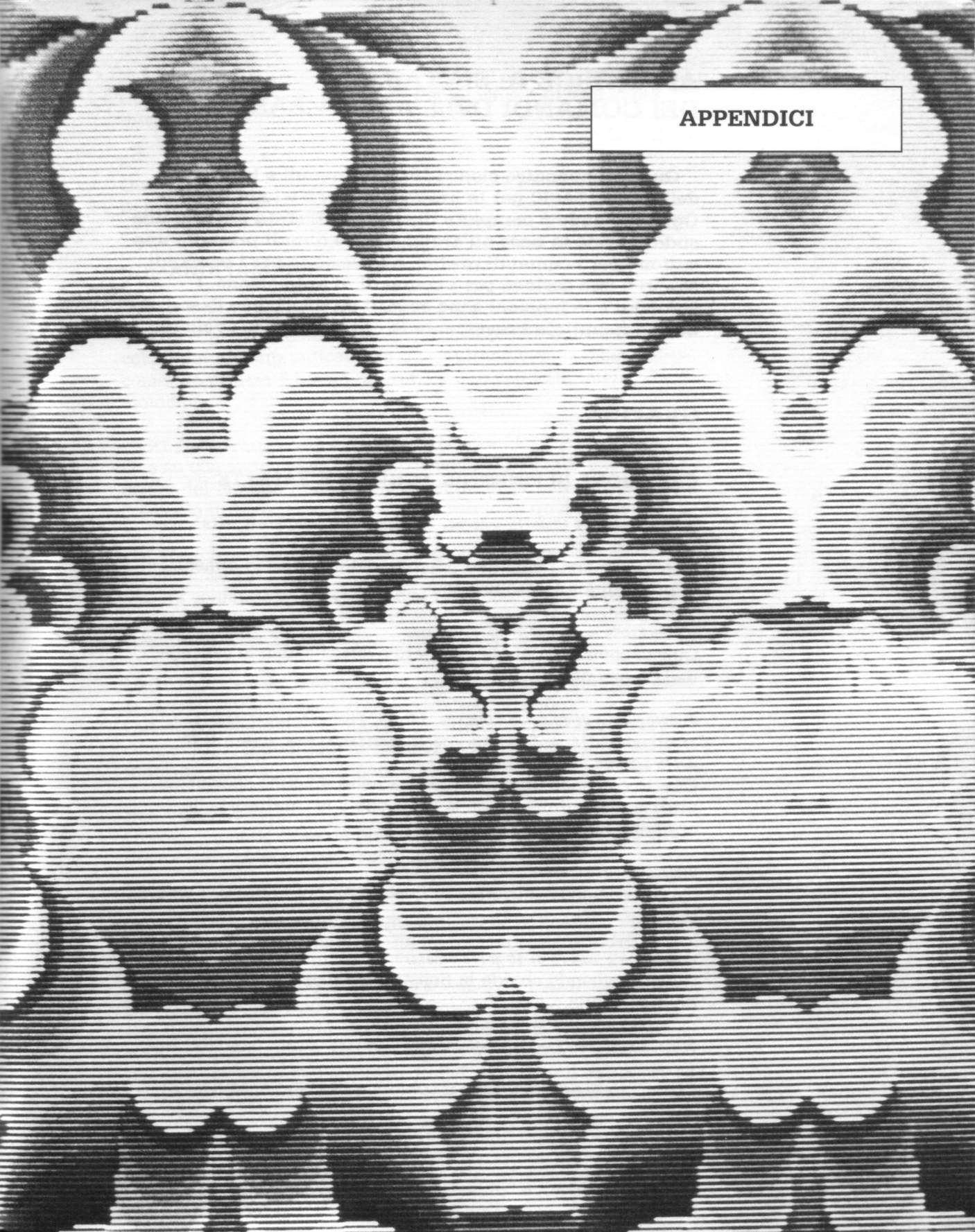
In molti casi, le enormi possibilità dell'Amiga non vengono pienamente sfruttate nemmeno nei programmi di applicazione. Ad esempio, lo scorrimento dello sfondo non è previsto né da Deluxe Video né da Aegis Animator. Sono completamente inutilizzate anche le routine del software sistema, che consentono all'operatore di creare una figura animata composta da più elementi e molto realistica nei movimenti. Ma i programmi di animazione sono un prodotto molto recente e, con l'intensificazione della concorrenza, assisterete sicuramente alla pubblicazione di programmi sempre più completi e raffinati.

I musicisti sono irresistibilmente attratti dall'Amiga, soprattutto dalle possibilità di campionamento dei suoni offerte dalla macchina. La stessa capacità dell'Amiga di svolgere più attività contemporaneamente consente di controllare e registrare tutto ciò che avviene su un'intera rete di strumenti MIDI. Molti di questi musicisti sono anche programmatori, quindi non dovrete attendere molto per la creazione di programmi che consentano di realizzare strumenti molto sofisticati, di trascri-

vere direttamente su pentagramma le vostre performance alla tastiera, oppure programmi didattici per l'apprendimento della notazione musicale.

È impossibile prevedere quali programmi verranno realizzati in futuro: gli utenti aumentano costantemente le loro richieste e le loro esigenze, e i programmatori non tardano mai ad accettare la sfida. I programmatori potrebbero arrivare tanto a fondo nella conoscenza dell'hardware e del software del sistema da scoprire possibilità che i creatori stessi dell'Amiga non hanno nemmeno immaginato.

Indipendentemente da tutto l'hardware e il software che potete aggiungere al vostro computer, la cosa forse più importante è riuscire a non dar loro troppo peso e dedicarsi a creazioni personali. Tutti i computer, per quanto affascinanti siano le possibilità offerte, non sono altro che semplici strumenti; e lo scopo di uno strumento è quello di trasformare in realtà le idee del suo possessore, nel modo più semplice e fedele possibile.



APPENDICI

APPENDICE A: SINTASSI DEI COMANDI DELL'AMIGA BASIC

Questa appendice elenca tutti i comandi e le funzioni dell'Amiga BASIC per la grafica, il suono e l'animazione. Ogni comando è completo di sintassi. È stata operata una suddivisione in base all'argomento: è quindi presente una sezione per i comandi grafici, una sezione per i comandi sonori e una sezione per i comandi di animazione. I comandi e le funzioni sono seguiti da una breve descrizione che ricapitola brevemente le operazioni che svolgono.

GRAFICA

CREAZIONE DI FINESTRE, SCHERMI E PALETTE

SCREEN *numero di ID dello schermo, larghezza, altezza, profondita', risoluzione*
Questo comando crea uno schermo definendone le dimensioni, la profondità e la risoluzione.

SCREEN CLOSE *numero di ID dello schermo*
Questo comando chiude lo schermo specificato e lo cancella dalla memoria.

WINDOW *numero di ID della finestra, titolo, (indirizzo dell'angolo) - (indirizzo dell'angolo), numero delle caratteristiche, numero di ID dello schermo*
Questo comando apre una finestra nello schermo specificato, utilizzando le dimensioni, la posizione e le caratteristiche opzionali indicate dall'utente.

WINDOW OUTPUT *numero di ID della finestra*
Questo comando trasforma in finestra output la finestra specificata.

WINDOW CLOSE *numero di ID della finestra*
Questo comando chiude la finestra specificata e la cancella dalla memoria.

PALETTE *numero del registro di colore, intensita' rosso, intensita' verde, intensita' blu*
Questo comando definisce il colore nel registro specificato

COLOR *numero del colore di primo piano, numero del colore di sfondo*
Questo comando fissa i colori di primo piano e di sfondo per tutti i comandi grafici successivi.

CLS
Questo comando cancella il contenuto della finestra output riempiendola del colore di sfondo.

COMANDI GRAFICI

PSET *(indirizzo), numero del registro di colore*
Questo comando colora il pixel specificato nell'indirizzo con il colore del registro indicato. Se quest'ultimo viene omesso, il comando utilizza il colore di primo piano valido in quel momento.

PRESET *(indirizzo), numero del registro di colore*
Questo comando colora il pixel specificato nell'indirizzo con il colore del registro indicato. Se quest'ultimo viene omesso, il comando utilizza il colore di sfondo valido in quel momento.

LINE *(indirizzo di partenza) - (indirizzo d'arrivo), numero del registro di colore, opzioni*
Questo comando traccia una linea retta che unisce il punto di partenza e quello d'arrivo, utilizzando il colore del registro specificato. Può anche creare riquadri pieni o vuoti.

CIRCLE *(indirizzo del centro), raggio, numero del registro di colore, punto di partenza dell'arco, punto di arrivo dell'arco, aspetto*
Questo comando traccia un cerchio intorno all'indirizzo del centro utilizzando il raggio specificato e il colore del registro indicato. Può anche disegnare archi ed ellissi.

AREA *(indirizzo)*
Questo comando fissa un punto all'interno di un poligono che deve essere colorato con il comando AREAFILL.

AREAFILL *modo*
Questo comando collega i punti indicati dal comando AREA e colora la superficie interna della figura che essi definiscono.

PAINT *(indirizzo), numero del registro di colore per l'area, numero del registro di colore per il perimetro*
Questo comando colora la superficie interna di un'area circoscritta utilizzando il colore del registro specificato e partendo dal pixel indicato nell'indirizzo. PAINT riconosce il perimetro se viene usato il colore del registro specificato nel comando.

PATTERN *maschera per le linee, nome array della maschera per le matrici grafiche*
Questo comando crea motivi decorativi utilizzati poi dai comandi che visualizzano figure sullo schermo.

MISCELLANEA DI COMANDI E FUNZIONI

PRINT *lista messaggi*
Questo comando stampa nella finestra output la stringa (le stringhe) di caratteri elencate nella lista.

LOCATE *numero di linea, numero di colonna*
Questo comando posiziona il cursore testo alla locazione specificata.

CSRLIN
Questa funzione restituisce il numero di linea della posizione del cursore testo.

POS(0)
Questa funzione restituisce il numero di colonna della posizione del cursore testo.

WINDOW *(numero di condizione)*
A seconda del numero specificato, questa funzione restituisce il numero di ID della finestra output o input, l'altezza, l'ampiezza e il numero di colori della finestra output, l'indirizzo del pixel all'interno della finestra output in cui si trova il cursore testo, oppure gli indirizzi di memoria utili ai programmatori più esperti.

POINT *(indirizzo del pixel)*
Questa funzione restituisce il numero del registro di colore del pixel specificato.

GET *(indirizzo dell'angolo) - (indirizzo dell'angolo), nome dell'array (numero dell'indice)*
Questo comando memorizza nell'array specificato i pixel contenuti nel rettangolo definito dagli indirizzi degli angoli opposti.

PUT *(indirizzo), nome dell'array (numero dell'indice), opzione*
Questo comando rimette sullo schermo all'indirizzo specificato il blocco di pixel immagazzinati da GET, combinandoli con i pixel già presenti sullo schermo in modo diverso a seconda dell'opzione specificata.

SUONO

MUSICA

BEEP
Questo comando produce un breve suono e fa lampeggiare lo schermo.

SOUND *frequenza, durata, volume, canale audio*
Questo comando suona una nota utilizzando le indicazioni dell'utente per la frequenza, il volume, la durata e il canale audio.

SOUND WAIT
Questo comando accoda tutti i comandi SOUND successivi senza suonarli.

SOUND RESUME
Questo comando permette l'esecuzione di tutti i comandi SOUND occupati da SOUND WAIT.

WAVE *canale audio, nome dell'array*
Questo comando crea una nuova forma d'onda nel canale specificato, utilizzando le informazioni contenute nell'array indicato dall'operatore.

SINTESI VOCALE

SAY *stringa di fonemi, nome dell'array di specificazione*
Questo comando converte la stringa di fonemi in forma udibile, utilizzando le caratteristiche definite nell'array di specificazione.

TRANSLATE$*(''stringa di testo'')*
Questa funzione trasforma la stringa in fonemi comprensibili per il comando SAY, seguendo la pronuncia inglese.

ANIMAZIONE

CREAZIONE E MOVIMENTO DEGLI OGGETTI

OBJECT.SHAPE *numero di ID dell'oggetto, stringa di definizione dell'oggetto*
Con questa sintassi il comando crea un oggetto in memoria in base ai dati contenuti nella stringa di definizione.

OBJECT.X *numero di ID dell'oggetto, coordinata x*
Questo comando fissa la coordinata x della locazione dell'oggetto specificato.

OBJECT.Y *numero di ID dell'oggetto, coordinata y*
Questo comando fissa la coordinata y della locazione dell'oggetto specificato.

OBJECT.ON *numero di ID dell'oggetto, numero di ID dell'oggetto, ...*
Questo comando visualizza nella finestra output l'oggetto specificato, o tutti gli oggetti nel caso l'utente non specifichi alcun numero di ID.

OBJECT.VX *numero di ID dell'oggetto, velocita' x*
Questo comando definisce la velocità x dell'oggetto specificato.

OBJECT.VY *numero di ID dell'oggetto, velocita' y*
Questo comando definisce la velocità y dell'oggetto specificato.

OBJECT.START *numero di ID dell'oggetto, numero di ID dell'oggetto, ...*
Questo comando mette in moto gli oggetti specificati, o tutti gli oggetti se il comando non è seguito da alcuna indicazione.

OBJECT.AX *numero di ID dell'oggetto, accelerazione x*
Questo comando fissa l'accelerazione dell'oggetto specificato lungo l'asse x (orizzontale).

OBJECT.AY *numero di ID dell'oggetto, accelerazione y*
Questo comando fissa l'accelerazione dell'oggetto specificato lungo l'asse y (verticale).

OBJECT.STOP *numero di ID dell'oggetto, numero di ID dell'oggetto, ...*
Questo comando blocca il movimento degli oggetti specificati, o di tutti gli oggetti se non è specificato nessun numero di ID.

OBJECT.OFF *numero di ID dell'oggetto, numero di ID dell'oggetto, ...*
Questo comando rende invisibili gli oggetti specificati, o tutti gli oggetti se l'utente non ne specifica nessuno in particolare.

OBJECT.CLOSE *numero di ID dell'oggetto, numero di ID dell'oggetto, ...*
Questo comando cancella dalla memoria l'oggetto specificato, o tutti gli oggetti se non viene precisato alcun numero di ID.

CONTROLLO DEGLI OGGETTI

OBJECT.SHAPE *numero di ID del duplicato, numero di ID dell'originale*
Con questa sintassi il comando OBJECT.SHAPE crea un duplicato di un oggetto già presente in memoria.

OBJECT.PLANES *numero di ID dell'oggetto, valore PlanePick, valore PlaneOnOff*
Questo comando cambia il colore di un oggetto esistente.

OBJECT.CLIP *(indirizzo dell'angolo) - (indirizzo dell'angolo)*
Questo comando definisce un rettangolo invisibile nella finestra output e lo utilizza come area entro cui è limitato il movimento degli oggetti.

OBJECT.X*(numero di ID dell'oggetto)*
Questa funzione restituisce la coordinata x della locazione dell'oggetto specificato.

OBJECT.Y*(numero di ID dell'oggetto)*
Questa funzione restituisce la coordinata y della locazione dell'oggetto specificato.

OBJECT.VX*(numero di ID dell'oggetto)*
Questa funzione restituisce la velocità x dell'oggetto specificato.

OBJECT.VY*(numero di ID dell'oggetto)*
Questa funzione restituisce la velocità y dell'oggetto specificato.

OBJECT.HIT *numero di ID dell'oggetto, valore della categoria, valore hit*
Questo comando definisce la categoria di un oggetto e le categorie di oggetti che esso può colpire.

OBJECT.PRIORITY *numero di ID dell'oggetto, valore di priorita'*
Questo comando assegna un valore di priorità all'oggetto specificato, valore che in caso di collisione fra due oggetti, determina quale sarà a passare sopra all'altro.

COLLISION ON
Questo comando attiva la lista delle collisioni.

ON COLLISION GOSUB *nome/numero della linea*
Questo comando fa saltare l'Amiga BASIC alla linea specificata ogni qualvolta si verifica una collisione.

COLLISION*(numero di ID)*
Questa funzione restituisce informazioni sulla lista delle collisioni.

COLLISION STOP
Questo comando sospende le operazioni della lista delle collisioni.

COLLISION OFF
Questo comando disattiva la lista delle collisioni e annulla l'effetto di ON COLLISION GOSUB.

SCROLL *(indirizzo dell'angolo) - (indirizzo dell'angolo), spostamento x, spostamento y*
Questo comando sposta il contenuto del rettangolo specificato nella direzione indicata da x e y.

APPENDICE B: CASE PRODUTTRICI CITATE NEL TESTO

Questa appendice elenca tutte le case produttrici menzionate nel testo, indicando per ciascuna l'indirizzo e il prodotto per cui è stata citata nel testo.

Aegis Development
2210 Wilshire #277
Santa Monica, CA 90403 USA
Prodotti: Aegis Animator, Aegis Images

A-Squared Systems
10 Skyway Lane
Oakland, CA 94619 USA
Prodotto: Amiga LIVE! framegrabber

Casio, Incorporated
15 Gardner Road
Fairfield, NJ 07006 USA
Prodotti: altoparlante AS-20, sintetizzatore CZ-101, sintetizzatore CZ-1000

Commodore International
1200 Wilson Drive
West Chester, PA 19380 USA
Prodotti: computer Amiga 500/1000/2000, Amiga 1300 Genlock Board, Amiga BASIC, AmigaDOS, Amiga Kickstart, Amiga Workbench, Graphicraft, Textcraft

Commodore Italiana S.p.A.
Via F.lli Gracchi, 48
20092 Cinisello Balsamo
Prodotti: distribuzione italiana dei prodotti CBM, interfacce Genlock A2995 e A2997, schede di espansione per A500 e A2000

Electronic Arts
1820 Gateway Drive
San Mateo, CA 94404 USA
Prodotti: Deluxe Music, Deluxe Paint, Deluxe Video, Instant Music

Epson America, Incorporated
2780 Lomita Boulevard
Torrance, CA 90505 USA
Prodotto: stampante JX-80

The Micro Forge
398 Grant Street, S.E.
Atlanta, GA 30312-2227 USA
Prodotto: schede RAM

Mimetics Corporation
P.O. Box 60238 Station A
Palo Alto, CA 94306 USA
Prodotti: SoundScape, campionatore SoundScape

NewTek Incorporated
701 Jackson Suite B3
Topeka, KS 66603 USA
Prodotto: DigiView

Okidata
532 Fellowship Road
Mt. Laurel, NJ 08054 USA
Prodotto: stampante Okimate 20

Pioneer Video, Incorporated
200 West Grand Avenue
Montvale, NJ 07645 USA
Prodotti: lettore dischi laser CLD-900, interfaccia per computer IU-04

Sony Corporation of America
Sony Drive
Park Ridge, NJ 07656 USA
Prodotto: monitor V-1311

Teac Corporation of America
7733 Telegraph Road
Montebello, CA 90640 USA
Prodotto: registratore TASCAM Ministudio Porta One

Video Vision Associates Limited
7 Waverly Place
Madison, NJ 07940 USA
Prodotto: dischi laser Space Archive

Xerox Corporation
901 Page Avenue
Fremont, CA 94538 USA
Prodotti: stampanti a getto d'inchiostro Diablo C-150 e Xerox 4020

INDICE ANALITICO

I numeri di pagina delle illustrazioni sono indicati in corsivo.

A

Accesso Diretto alla Memoria (DMA), 14
Aegis Animator, 284, 311-12, 375
Altoparlanti, 196-97
Amiga BASIC, 23, 49, 52, 375
Amiga BASIC, animazione, 318-50. *Vedi anche* Object Editor.
  campi visivi, 283-84, 324
  cassetti per immagazzinare i file, 331-32
  controllo del movimento, 352-76 (*vedi anche* Appendice A; *comandi e funzioni specifiche*)
  creazione di un'area limitata di movimento, 352-53. (*vedi anche* comando OBJECT.CLIP).
  gel, 284
  gestione delle collisioni, 358-68 (*vedi anche* Appendice A; *comandi e funzioni specifiche*)
  lettura della locazione e della velocità di un oggetto, 354-57, *357* (*vedi anche funzioni specifiche*).
  posizionamento e movimento degli oggetti, 329-42 (*vedi anche comandi specifici*)
  scorrimento, *283*
  trucchi, 368-74
Amiga BASIC, grafica, 96-186. *Vedi anche* Appendice A; palette di colori; schermo; finestre
  creazione di immagini, 118-24 (*vedi anche* cursore grafico e indirizzamento dei pixel).
  finestra dello schermo del Workbench, 183
  finestre list e output, 96-100
  grafica e funzioni miste, 160-86 (*vedi anche* Appendice A; indirizzamento dei caratteri; fonti; grafica, copiatura e reinserimento; grafica, interattiva)
  registri di colore
Amiga BASIC, suono, 240-68. *Vedi anche* Appendice A, *comandi Sound specifici;* Sintesi vocale, generazione
  array della scala musicale, 252-53
  esecuzione di una partitura musicale, 254-57
  sincronizzazione, 245-47
AmigaDOS (Disk Operating System), 26, 29
  comandi CLI, 29
  Exec, 25
  RAM, 26
Ampiezza (volume), 193, 204, 206
Amplificatore, 198
Animazione, 272-350. *Vedi anche* Deluxe Video
  interattiva, 282
  moto interno, 286-87
  principi fondamentali, 272-82 (*vedi anche* Movimento illusorio; Prospettiva
  standard IFF, 313
Anti-aliasing, 46
AREA, comando, 143
AREAFILL, comando, 144-46
  riempimento di un'area con una matrice grafica, 146
Array, forma d'onda. *Vedi* Forma d'onda, tavola
Array, scala musicale, 252-53
  notazione musicale tradizionale, *253*
Array, specificazione, 261-66. *Vedi anche* SAY, comando
  scelta dell'assegnazione del canale, 264-65
  selezione dell'inflessione o del monotòno, 262
  selezione delle opzioni per più comandi SAY, 265-66
  selezione del discorso sincrono o asincrono, 265
  selezione del tipo di voce, 263
  regolazione del tono di base, 262
  regolazione della frequenza di campionamento, 263
  regolazione della velocità di riproduzione, 263
  regolazione del volume, 264
Array, variabile PUT e GET, 176-81. *Vedi anche* PUT e GET, comandi
  calcolo delle dimensioni, 178-79
  copia di più blocchi grafici di diverse dimensioni in un unico array, 180-81
  memorizzazione di un blocco grafico, 179
  memorizzazione di dati grafici, 177-78, *178*
ASCII (American Standard Code for Information Interchange), 211

B

BEEP, comando, 240
  forma d'onda di, 240
Bit-plane, 65, 106-07
Blend, modo, 73, 79
  creazione di nuvole, *80*
Blitter, 55
Bob, 323
Brush, modi, Deluxe Paint, 71-82
  blend, 73

blend per creare nuvole, 79
color, 72
cycle, 74
object, 72
object e color per contornare le figure, 74-76
pennello testo per la realizzazione di scritte contornate, *76*
replace, 72
shade, 73
shade per creare ombre, 80-82
smear, 73
smear per creare uno sfondo sabbioso, 76-78

C

Campo visivo, 291-96
Campo visivo, animazione, 324-28
profondità dello schermo, 325-26
risoluzione dello schermo, 324-25
scelta dei colori, 326-27
Case produttrici (menzionate nel testo), 385
Chip della memoria, 13. *Vedi anche* RAM; ROM
Ciclo infinito, 135
CIRCLE, comando, 138-43
cerchio misurato in radianti, 141
creazione di archi, 139-41, *140*
creazione di cerchi, 138-39
creazione di ellissi, 141-43
CLI (Command Line Interpreter), 28-29, 84
CLS, comando, 132
COLLISION, funzione, 364-66
lista delle collisioni, 364, *364*
COLLISION OFF, comando, 368
COLLISION ON, comando, 362
COLLISION STOP, comando, 368
Colore, 34-37
creazione di colori RGB, 35
HIS (Tonalità, intensità, saturazione), 36-37
primario, 35
COLOR, comando, 120-21
memorizzazione del colore, 121-22
insieme al comando PRINT, 167-68
modifica del colore di sfondo, 120
modifica del colore di primo piano, 120
Color, modo, 72, 74-76
Comandi. *Vedi* Appendice A *e comandi specifici*
Commodore-Amiga, 89, 90
Convertitore di segnali video, 16
Cursore grafico e indirizzamento dei pixel, 126-57 *Vedi anche comandi grafici specifici;* Creazione di figure poligonali
indirizzamento assoluto, 127
indirizzamento relativo, 127-29
margini delle finestre, 129-31
Cursore testo
posizionamento con le funzioni CSRLIN e POS(0), 166-67
Cycle, modo, 74

D

Deluxe Music, 29, 214-230
caratteristiche, 216-25
consigli, 227
effetti speciali con il registratore Tascam Ministudio Porta One, 229-30
registrazioni, 228-30
sincronizzazione delle partiture, 230
uso del MIDI, 225-27
Deluxe Paint. *Vedi anche* Brush, modi; Palette di colori
creazione di disegni precisi e animazione a ciclo di colori, 295-300
creazione di immagini in sequenza per animazioni, 300-06
creazione di sfondi con oggetti, 291-95
creazione di un disco di lavoro, 63-64
riproduzione fotografica, 87-88
riproduzione su carta, 83-87
selezione della risoluzione e della profondità dello schermo, 64-65
trucchi, 82-83
Deluxe Video, 29, 52, 284
e Deluxe Paint, 291-306 (*vedi anche* Deluxe Paint)
blitter, 290, 295
ciclo di colori, 295-300, *296, 297, 298, 300*
dischi library, 312
disegno per sequenze, 300-06, *301, 302, 303, 304*
scrittura di copioni leggibili, 307-08
sincronizzazione con il suono, 306
videoregistrazione, 308-10
Deluxe Video Framer, 300, 301, 303
Deluxe Video Maker, 300, 305
Denise (coprocessore), 284
Disegni in sequenza, 286
Disegni precisi, 68-71
allineamento della griglia, 69-71
assegnazione delle coordinate e posizionamento del pennello, 68
tasto SHIFT per tracciare linee perfettamente rette, 69
blitter
Disk drive esterno, *21*
Duplicazione di oggetti. *Vedi* Object.Shape, comando
Durata, 203, 204e comando SOUND, 243, 254

E

Electronic Arts, 230, 290, 312
Esadecimali, numeri, 151-52
Espansione Commodore A501
    Commodore A2052, 314
Exec, 25
    e il Kickstart, 25
    e la ROM, 25

F

Fat Agnus, *11*
Figure poligonali, creazione, 143-47, *147*. *Vedi anche* AREA, comando; AREAFILL, comando.
Finestra Output, 116-17
Finestre. *Vedi anche* Finestra Output; WINDOW CLOSE, comando; WINDOW OUTPUT, comando; WINDOW, comando
Firmware, 13
Fonti, utilizzazione di diverse dimensioni, 161-62
Forma d'onda
Frequenza (dei suoni), 192-93
    e comando SOUND, 241-43
Funzioni. *Vedi anche* Appendice A *e funzioni specifiche*

G

Gadget di dimensionamento, 131
Gel, 284-88
    bob, 285-86
    sprite, 284-85
GET, comando, 176, 184-86
Grafica, copiatura e reinserimento, 176-86. *Vedi anche* Array, variabili PUT e GET; GET, comando; PUT, comando
    esempi dimostrativi di GET e PUT, 184-86
    caratteristiche grafiche, 48-60. *Vedi anche* Palette dei colori; disegno di immagini; Fusione di due immagini; Dettagli delle immagini; Scorrimento; Testo, creazione
    grafica interattiva,169-76. *Vedi anche* POINT, funzione; WINDOW(), funzione
Graphicraft, 283

H

Hardware, 4-22, *4*. *Vedi anche* Unità centrale; Tastiera; Monitor; Mouse
    fabbricanti (menzionati nel testo), 385
Hardware, animazione, 313-16
    lettori di dischi laser, 315-16
    scheda Amiga 1300 Genlock, 314
    schede di espansione RAM, 313-14
Hardware, grafica, 90-94. *Vedi anche* Monitor; Stampanti
    Amiga Live! framegrabber, 93
Hardware, suono
    altoparlanti, 234-35
    stereo, 234
Hawkins, Trip, XI

I

Immagini, creazione, 55-56
    blitter, 55
    e coprocessore Agnus, 55
    fusione di due immagini, 59, *59*
    dettagli delle immagini, 49-53. *Vedi anche* Risoluzione
Indirizzamento dei caratteri, 160-61
    numerazione delle linee e delle colonne, *161*
Indirizzamento dei pixel. *Vedi* Cursore grafico e indirizzamento dei pixel.
Interfaccia multistandard Genlock A2997, 315
    VHS/BETA A2998, 315
Interlacing, 105
Intuition, 26-27, 52
    routine, 26

L

Libraries e devices, 25-26, 283
LINE, comando, 136-38
    creazione di riquadri, 137-38, *138*
    creazione di linee, 136-37, *137*
LOCATE, comando, 164-65

M

Maschera lineare binaria, 151-52
MIDI (Musical Instrument Digital Interface), 208-11, *210*
    interfaccia, 236-37
    creazione di un circuito MIDI, 210
    messaggi, 211
    standard, 209-10
Mixer, 207-08
Modem, 18, 22, 211
Modulatore TV, 19
Monitor, 6, 38-48, 90-91
    a colori, 40-42, *42*
    compositi, 6
    fosfori di un monitor a colori, 41
    monocromatici, 38-39
    porte, 6
    proiettore di elettroni, 39, *39*
    RGB, 6, 90
    scansione del quadro, 39-40
    televisione, 6
    trasferimento di un'immagine video su altri apparecchi, 42-48 (*vedi anche Riproduzione fotogra-*

fica dello schermo; Riproduzione su carta delle immagini video; Videocassetta)
tubo catodico, 39
Monitor RGB, 6, 90-91
analogici, 90-91
digitali, 90-91
Moto dei componenti, 287
Moto illusorio, 272-76, *273*
oggetti animati, 275-76
sfondo, 274-75
Mouse, 8
Musica sintetizzata, 199-205. *Vedi anche* Sequencer; Sintetizzatori

O

OBJECT.AX, comando, 339-40
OBJECT.AY, comando, 339-40
OBJECT.CLIP, comando, 353-54
OBJECT.CLOSE, comando, 342
Object Editor, 318-24
disegno di un oggetto, 321-24
modifica dell'oggetto, 319-21
OBJECT.HIT, comando, 358-61
Object, modo, 72
OBJECT.OFF, comando, 341-42
OBJECT.ON, comando, 334-35
OBJECT.PLANES, comando, 345-50
esempio, 350
per cambiare i colori, 345-47
regolazione dei valori PlanePick e PlaneOnOff, 347-49
OBJECT.PRIORITY, comando, 361-62
OBJECT.SHAPE, comando, 330, 331
per duplicare oggetti, 343-44
OBJECT.START, comando, 338-39
OBJECT.STOP, comando, 341
OBJECT.VX(), funzione, 355-56
OBJECT.VX, comando, 336-38
OBJECT.VY(), funzione, 355-56
OBJECT.VY, comando, 336-38
OBJECT.X(), funzione, 354-55
OBJECT.X, comando, 332-34
OBJECT.Y(), funzione, 354-55
OBJECT.Y, comando, 332-34
ON COLLISION GOSUB, comando, 363
Oscillatore controllato in tensione, 200

P

PAINT, comando, 147-50
per riempire la superficie interna di un cerchio, 149, *150*
PALETTE, comando, 121-24
creazione di palette diverse per schermi diversi, 123-24
due schermi diversi con due palette esclusive, *124*
16 intensità di colori primari e corrispondenti valori PALETTE, *122*
Palette di colori, 53-55, 118-24
creazione, 66-68
finestra della palette, *66*
gamma, 67
HAM (Hold and Modify), 54
immagini che utilizzano il modo HAM per le sfumature, 55
Partitura musicale, esecuzione. *Vedi* Amiga BASIC, suono
Pattern area, 153-57
applicazione di una pattern area, 154-57
memorizzazione dell'area in un array integrale, 153-54
PATTERN, comando, 150-57. *Vedi anche* Pattern area; Pattern linea
applicazione di una pattern linea, 152-53
conversione di una maschera lineare binaria in un numero esadecimale, 151-52
creazione di una pattern linea, 151-53
modifica dei colori della matrice grafica, 157
Pattern linea, 151-53
applicazione di una pattern linea, 152-53
conversione di una maschera lineare binaria in un numero esadecimale, 151-52
Periferiche
disk drive esterni, 21-22
RAM esterna, 22
stampanti a getto d'inchiostro, 20
stampanti a impatto, 20
stampanti a margherita, 20
stampanti laser, 21
stampanti termiche, 20
Pioneer CLD-900, 315-16
Pioneer IU-04, interfaccia, 316
POINT, funzione, 175-76
Porte e connettori. *Vedi anche*
Periferiche audio, 18
connettore della RAM interna, 14-15
connettore per l'espansione, 15-16
di controllo, 13
disk drive, 18
MIDI, 209
modulatore TV, 19
parallela, 17
RGB, 19
sul lato destro e anteriore dell' unità centrale, *14*
sul retro dell'unità centrale, *17*
seriale, 18
tastiera, 17
video, 19
Preferences, 60, 84-87
controlli RGB nel programma Preferences, *36*

schermo Change Printer, 84-86, 87, 85, 86
schermo Graphic Select, 86, 87
PRESET, comando, 123
PRINT, comando, 163
insieme al comando COLOR, 167-68
Prospettiva, 276-81
a due dimensioni, 277, *277*
a due dimensioni e mezzo, 280-81, *281*
a tre dimensioni, 278-80, 278
PSET, comando, 132-33
con formule trigonometriche, 134-35, *135*
PUT, comando, 181-82
animazione sequenziale, 371-74, *373*
definizione dell'indirizzo di PUT, 181
nome dell'array e indirizzo, 181-82
opzioni, 182-84

R

RAM (Random Access Memory), 13
bit-plane, 65
fonti, 57
memorizzazione delle animazioni, 313-14
refreshing, 113-14
risoluzioni, 52
sequencer, 199
Workbench, 27
Registratori a cassette, 207
Registrazione, animazione, 308-10
qualità, 310
Registrazione, partiture di Deluxe Music, 228-29
Registrazione, suono, 206-08
Replace, modo, 72
Riproduzione fotografica dello schermo, 43-44
svantaggi, 44
Riproduzione su carta delle immagini di Deluxe Paint. *Vedi anche* Preferences, 84-87
Riproduzione su carta di immagini video, 44-48
conversione di immagini a colori in riproduzioni in bianco e nero, 46-47
fedeltà, 44-46
riproduzione a colori, 47-48
Risoluzione
animazione, 324-25
anti-aliasing, 46, *46*
comando SCREEN, 104-05
dimensione delle fonti, 162
immagini che utilizzano pixel ad alta risoluzione, *51*
immagini che utilizzano pixel a bassa risoluzione, *50*
influenza sulla memoria, 52
mista, 52
profondità, 64-65
"seghettature", 44-47, *46*
visualizzazioni del Workbench che utilizzano lo schermo a risoluzione 640 per 200, *50*
ROM (Read Only Memory), 13, 24

S

SAY, comando, 260-68. *Vedi anche* Array, specificazione
codice di fonemi, 261
esempi, 266-68
Scheda IBM XT, *11*
Schermi, 100-01, 104-09. *Vedi anche* SCREEN CLOSE, comando; SCREEN, comando
linea d'intestazione, 100, 101
Schermo (aree orizzontali), 52
Scorrimento, 58, *58*
SCREEN, comando, 104-09
cinque schermi simultanei, 108, *108*
conservazione della memoria, 107
esempi, 107-09
larghezza e altezza, 105-06
memoria richiesta, 107
numero di ID, 104
profondità, 106-07
schermo modo 1, 106, *106*
SCREEN CLOSE, comando, 109
SCROLL, comando, 369
scorrimento del campo visivo, 369-70
Sequencer, 205
Shade, modo, 73, 80-82
creazione di ombre, *82*
Sintesi vocale, 206
Sintesi vocale, generazione, 257-68 (*vedi anche* SAY, comando; TRANSLATE$(), funzione
Sintetizzatore Casio CZ-101, 237
Sintetizzatore Casio CZ-1000, 237
Sintetizzatori, 199-203
analogici, 200
convertitore digitale-analogico, 202
digitali, 200-03
Smear, modo, 73, 76-78
creazione di uno sfondo sabbioso, *78*
Software, 23-29, *24*. *Vedi anche* AmigaDOS; CLI (Command Line Interpreter); Exec; Intuition; Libraries e devices; Workbench
Software, animazione. *Vedi* Aegis Animator; Amiga BASIC, animazione; Deluxe Video
Software, grafica, 88-90. *Vedi anche*

Amiga BASIC; Deluxe Paint
Aegis Images, 89
Graphicraft, 89
standard grafico IFF (Interchange File Format), 89-90
Software, suono. *Vedi* Amiga BASIC, suono; Deluxe Music
Instant Music, 231
standard musicale IFF, 233
SOUND WAIT, comando
SOUND RESUME, comando, 245-47
SOUND, comando, 241-45
regolazione della durata, 243
regolazione della frequenza, 241-43, *242*
regolazione del volume, 244
scelta del canale audio, 244-45
SOUND WAIT, comando, 245-47
Suoni elettronici, 196-99
Stampa
driver della stampante, 60, 84
Stampanti, 20-21, 91-93
Diablo C-150 e Xerox 4020, 92-33
Epson JX-80, 92
Okimate 20, 91-92
Stampanti seriali, 18
Stereofonia, 198-99
Suono. *Vedi anche* MIDI; Registrazione; Suono, elettronico; Musica sintetizzata; Sintesi vocale
campionato, 203-05
campionatori, 236
caratteristiche, 192-95
onde sinusoidali, quadrate e a dente di sega, *194*
ricezione dell'orecchio, 190-92
Suono, sincronizzazione. *Vedi* Amiga BASIC, suono; SOUND RESUME, comando

T

Tastiera, 7-8, *7*
Tavola della forma d'onda, 248-51
assegnazione dei dati della tavola a un array, 249
assegnazione di un canale audio, 251
calcolo, 249
creazione, 248-49
Testo, creazione, 56-57, *56*
fonte, 56-57
sfondo, 57
Textcraft, 57
Timbro, 193-94, 204, 205, 240
Tono, 204, 206, 254
TRANSLATE$(), funzione, 258-60
compitazione alternativa per una pronuncia corretta, 260
uso della punteggiatura, 259-60
Trigonometria, 134-35

U

Unità centrale, 8-14
apertura dell'unità, 16
chip della memoria, 13
coprocessori, 12-13
disk drive interno, 13-14
microprocessore Motorola 68000, 12
scheda madre, 10
trasformatore, 9

V

Videocassetta, 43
Videodigitalizzatore DIGIVIEW, 94
Video Vision Associates, dischi laser Space Archive, 316
Volume, 254
e comando SOUND, 244

W

WAVE, comando, 247-51. *Vedi anche* Tavola della forma d'onda
WINDOW CLOSE, comando, 118
WINDOW(), funzione, 170-175
esempi, 171-75, *172, 174, 175*
tabella, *171*
WINDOW, comando, 105-16
caratteristiche, 112-14
esempi, 115-16
finestra con tutte le caratteristiche possibili, *114*
finestra nell'angolo superiore sinistro dello schermo, *112*
indirizzi degli angoli opposti, 110-12
numero di ID, 110
numero di ID dello schermo, 115
sistema di indirizzamento dei pixel, *111*
WINDOW OUTPUT, comando, 117-18
Workbench, 27-28, 56, 60, *27*
e Intuition, 28

L'autore

Suonatore di oboe, Michael Boom entrò nel mondo dei computer nel 1980, quando acquistò un Atari 800. È l'autore di *Understanding Atari Graphics, How to Use Atari Computers* e *How to Use the Commodore 64*, tutte opere pubblicate come manuali della Alfred Handy. Ha scritto anche per la rivista *Compute!*. Negli ultimi due anni ha lavorato come consulente nel campo dei microcomputer, e per un anno è stato consulente musicale della Commodore-Amiga. Michael Boom vive attualmente a Oakland, in California, negli Stati Uniti.

Questo volume è stato stampato nel mese di gennaio 1988
presso gli stabilimenti della Rotolito Lombarda S.p.A.
Stampato in Italia – Printed in Italy

L'AMIGA

Immagini, Suoni e Animazioni sul Commodore Amiga

Michael Boom

L'apparizione del Commodore Amiga ha suscitato nel mondo dei computer tanta eccitazione quanta non se ne vedeva da anni. Questo volume, scritto da un consulente di primo piano della Commodore-Amiga, susciterà altrettanta creatività in tutti coloro che si preparano a usare o ad acquistare questa macchina favolosa.

Se siete appassionati di grafica, se vi interessano le realizzazioni audio e video, se vi piacciono i computer e conoscete un minimo di BASIC, preparatevi a raggiungere col vostro Amiga 500/1000/2000 risultati incredibili, che andranno ben al di là di un livello da principiante. Questo volume vi insegnerà a:

- Esplorare le frontiere dell'immagine più sofisticata, imparando a sviluppare tutta la potenza dell'Object Editor.
- Produrre suoni reali e sintetizzati con il processore interno dell'Amiga.
- Creare sequenze animate e registrarle su videocassetta.

Oltre a una quantità di informazioni su come ottenere il meglio dall'Amiga BASIC, imparerete tecniche avanzate e istruzioni per l'uso di importanti pacchetti software come Deluxe Paint, Deluxe Music e Deluxe Video. *L'Amiga* sarà la guida indispensabile per avventurarvi nella miniera d'arte e di possibilità che offre il vostro fantastico computer.

ISBN 88-7803-000-7

9 788878 030008

Lire 60.000 (i.i.)

L'AMIGA

Michael Boom